



AFRL-RI-RS-TR-2014-153

## **MULTIMEDIA-BASED INTEGRATION OF CROSS-LAYER TECHNIQUES**

---

SAN DIEGO STATE UNIVERSITY

*JUNE 2014*

FINAL TECHNICAL REPORT

***APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED***

STINFO COPY

**AIR FORCE RESEARCH LABORATORY  
INFORMATION DIRECTORATE**

## **NOTICE AND SIGNATURE PAGE**

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RI-RS-TR-2014-153 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

**/ S /**

MICHAEL J. MEDLEY  
Work Unit Manager

**/ S /**

MARK H. LINDERMAN  
Technical Advisor, Computing &  
Communications Division  
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p><b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b></p>					
1. REPORT DATE (DD-MM-YYYY) JUNE 2014		2. REPORT TYPE FINAL TECHNICAL REPORT		3. DATES COVERED (From - To) OCT 2010 – MAR 2014	
4. TITLE AND SUBTITLE  MULTIMEDIA-BASED INTEGRATION OF CROSS-LAYER TECHNIQUES				5a. CONTRACT NUMBER FA8750-11-1-0048	
				5b. GRANT NUMBER N/A	
				5c. PROGRAM ELEMENT NUMBER 62702F	
6. AUTHOR(S)  Sunil Kumar				5d. PROJECT NUMBER AN11	
				5e. TASK NUMBER SD	
				5f. WORK UNIT NUMBER SK	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) San Diego State University 5500 Campanile Drive, MC 1901 San Diego, CA 92182-0413				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  Air Force Research Laboratory/RITE 525 Brooks Road Rome NY 13441-4505				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RI	
				11. SPONSOR/MONITOR'S REPORT NUMBER AFRL-RI-RS-TR-2014-153	
12. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT This report discusses the design of cross-layer protocols for the transmission of delay-sensitive and prioritized data in wireless networks; these protocols consider the QoS issues in an end-to-end fashion and collaboratively design protocols at different network layers. First, a novel cross-layer scheme is discussed which minimizes the expected received video distortion by jointly optimizing the packet sizes at the application (APP) layer and estimating their forward error correction (FEC) code rates to be allocated at the physical (PHY) layer for bit-rate limited and noisy channels. Second, a cross-layer FEC scheme is discussed, which jointly optimizes the Raptor codes at APP layer and rate compatible punctured convolutional (RCPC) codes at PHY layer for the prioritized video packets, in order to minimize the distortion for the given source bit rates and channel constraints. Finally, a video slice CMSE and deadline-aware sliding-window based scheduling algorithm is designed, which exploits the temporal and SNR scalability of a H.264/SVC compressed bit stream for transmission over a wireless link with time-varying bit rate.					
15. SUBJECT TERMS Cross-layer, wireless networks, video transmission, QoS, H.264/AVC, packet size adaptation, scheduling, error correcting codes, optimization spectra and dynamic spectrum allocation					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT  UU	18. NUMBER OF PAGES  160	19a. NAME OF RESPONSIBLE PERSON MICHAEL J MEDLEY
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) N/A

# TABLE OF CONTENTS

<b>LIST OF FIGURES</b> .....	v
<b>LIST OF TABLES</b> .....	viii
<b>SUMMARY</b> .....	1
<b>1.0 INTRODUCTION</b> .....	3
1.1 Motivation.....	3
1.2 Objectives.....	4
1.3 Organization of Report.....	4
<b>2.0 BACKGROUND AND ASSUMPTIONS</b> .....	7
2.1 Modeling the Impact of other Layers on Cross-Layer Protocols.....	10
2.2 Design of Cross-Layer Rate Control, Payload Adaptation, Packet Scheduling and FEC Protocols.....	11
<b>3.0 CROSS-LAYER PRIORITY-ADAPTIVE PACKETIZATION AND ERROR CORRECTION FOR WIRELESS CHANNELS</b> .....	13
3.1 Introduction.....	13
3.2 Related Work.....	14
3.3 Methods, Assumptions and Procedures.....	16
3.3.1 Proposed Cross-Layer Approach.....	16
3.3.1.1 CMSE Computation/Prediction of H.264 Video Slices.....	17
3.3.1.2 H.264 Video Packet Formation.....	18
3.3.2 Expected Video Distortion Minimization.....	19
3.3.2.1 Packet Formation (PF) Block.....	20

3.3.2.2 Distortion Minimization with OCRA Block.....	22
3.3.2.3 Determination of $\lambda$ .....	27
3.3.2.4 Discarding Packets.....	28
3.3.3 Frame-Level DP-UEP Scheme.....	28
3.3.4 Frame-Level DP-UEP Using Prediction.....	31
3.3.4.1 GLM Approach for Estimating $w_{ovh}^l$ .....	32
3.3.4.2 Response Variable Distribution.....	33
3.3.4.3 Model Fitting and Validation.....	35
3.3.5 Problem Formulation of other Error Protection Schemes.....	38
3.4 Results and Discussion.....	39
3.4.1 Simulation Setup.....	40
3.4.2 Performance of DP-UEP Scheme.....	41
3.4.3 Performance of DP-UEP(frame) Scheme.....	46
3.5 Conclusion.....	49
<b>4.0 CROSS-LAYER FEC SCHEME FOR PRIORITIZED VIDEO TRANSMISSION OVER WIRELESS CHANELS .....</b>	<b>51</b>
4.1 Introduction.....	51
4.2 Related Work.....	52
4.3 Methods, Assumptions and Procedures.....	55
4.3.1 Cross-Layer UEP using FEC Codes for Video Transmission.....	55

4.3.1.1	Priority Assignment for H.264 Video Slices.....	55
4.3.1.2	Design of UEP Raptor Codes at APP.....	57
4.3.1.3	Design of RCPC Codes at PHY.....	60
4.3.1.4	System Model at Transmitter.....	63
4.3.1.5	Decoding at Receiver.....	65
4.3.2	Cross-Layer Optimization of FEC Codes.....	65
4.3.2.1	Formulation of Optimization Problem.....	66
4.4	Results and Discussion.....	69
4.4.1	Discussion of Cross-Layer Optimization Results.....	72
4.4.2	Performance of Cross-Layer FEC Schemes for Test Videos over AWGN Channels.....	79
4.4.3	Performance of Cross-Layer FEC Schemes for Test Videos over Fading Channels .....	85
4.5	Conclusions.....	88
<b>5.0</b>	<b>CROSS-LAYER SCHEDULING SCHEME FOR VIDEO TRANSMISSION OVER WIRELESS NETWORKS .....</b>	<b>90</b>
5.1	Introduction.....	90
5.2	Related Work.....	93
5.3	Methods, Assumptions and Procedures.....	98
5.3.1	System Model.....	98
5.3.1.1	Scalable Video Coding.....	98
5.3.1.2	Video Streaming System.....	101

5.3.1.3 Wireless Channel.....	105
5.3.2 Problem Formulation.....	106
5.3.2.1 EDF-based Scheme.....	106
5.3.2.2 CMSE-based Scheme.....	107
5.3.2.3 Proposed Scheme.....	112
5.4 Results and Discussion.....	115
5.4.1 Simulation Setup.....	115
5.4.2 Evaluation of Average Goodput and Percentage of Expired Whole Frames.....	119
5.4.3 Evaluation of Expired NAL Units.....	123
5.4.4 Evaluation of Video Quality.....	128
5.5 Conclusion.....	133
<b>6.0 CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS.....</b>	<b>134</b>
6.1 Conclusions.....	134
6.2 Contributions.....	135
6.3 Future Research and Recommendations.....	136
<b>7.0 REFERENCES.....</b>	<b>137</b>
<b>LIST OF ACRONYMS.....</b>	<b>147</b>

## LIST OF FIGURES

Figure 1: Flow diagram of proposed cross-layer system .....	16
Figure 2: Block diagram of proposed dynamic programming approach.....	20
Figure 3: Packet formation in PF block.....	22
Figure 4: Cumulative distribution function (CDF) for the binned observations and fitted distributions of $w_{ovh}^l$ .....	35
Figure 5: Average video PSNR (dB) and corresponding average VQM comparison computed over 100 realizations of each AWGN channel for Foreman:(a),(b), and Silent:(c),(d).....	42
Figure 6: Average number of slices discarded per GOP in EEP-slice-ENH, Dual15, and DP-UEP for Foreman.....	44
Figure 7: Distribution of the final output bits for Foreman at 3 dB channel SNR in EEP-slice-ENH, Dual15 and DP-UEP schemes.....	45
Figure 8: Average video PSNR (dB) and average VQM comparison computed over 100 realizations of each AWGN channel for Akiyo: (a),(d), Foreman: (b),(e) and Stefan: (c), (f). The error-free PSNR values are: 46.5 dB for Akiyo, 37.3 for Foreman, and 29.7 for Stefan.....	48
Figure 9: The framework of our proposed Raptor encoder.....	59
Figure 10: Illustration of four cross-layer FEC schemes.....	64
Figure 11: A frame of three test video sequences.....	70



Figure 12: Normalized  $\bar{F}$  of Bus sequence for AWGN channel SNRs at channel bit rates:

(a)  $C = 1.4\text{Mbps}$  and (b)  $C = 1.8\text{Mbps}$ .....76

Figure 13: Average PSNR of test videos for different channel SNRs for AWGN channel: (a) Bus, (b) Coastguard, (c) Foreman sequence at  $C=1.4\text{Mbps}$ , and (d) Bus, (e) Coastguard, (f) Foreman at  $C=1.8\text{Mbps}$ . The PSNR of Bus, Coastguard, and Foreman at error-free channel are 30.24dB, 32.05dB, and 36.81dB, respectively.....80

Figure 14: Normalized  $F$  and average PSNR of test videos for channel SNRs at  $C = 1.4\text{Mbps}$  in Rayleigh flat fading channels with  $f_m = 41.7$ ,  $f_c = 900\text{MHz}$ , and speed of  $5\text{km/h}$ .....83

Figure 15: Normalized  $F$  and average PSNR of test videos for channel SNRs at  $C = 1.4\text{Mbps}$  in Rayleigh flat fading channels with  $f_m=41.7$ ,  $f_c=900\text{MHz}$ , and speed of  $50\text{km/h}$ .....84

Figure 16: Average PSNR of the optimal and sub-optimal FEC scheme (S-IV) for Akiyo over Rayleigh flat fading channel with  $f_m = 41.7$ ,  $f_c = 900\text{MHz}$  at speed of  $5\text{km/h}$ .....87

Figure 17: (a) Hierarchical prediction structure, and (b) motion-compensated prediction for MGS layers with key pictures.....99

Figure 18: Average R-D characteristic curves in terms of (a) CMSE, and (b) IMSE for different temporal and quality layers.....101

Figure 19: Video streaming system.....102

Figure 20: (a) Sampling curve for Table Tennis  $R_v(t)$ , and outgoing video bits supported by the channel, (b) close up of the sampling curve between  $t=15$  sec and  $t=16$  sec.....103

Figure 21: Video streaming timing diagram.....105

Figure 22: Sample iterations of our proposed dynamic programming algorithm over a window of frames at the video server. ....	114
Figure 23: Average goodput of the EDF-based, CMSE-based, and proposed scheduling schemes.....	120
Figure 24: Percentage of expired whole frames in EDF-based, CMSE-based, and proposed schemes over 120 random channel realizations.....	121
Figure 25: Percentage of expired whole frames in different temporal layers of EDF-based, CMSE-based, and proposed schemes over 120 random channel realizations.....	122
Figure 26: Total percentage of expired NAL units in 120 random channel realizations for EDF-based, CMSE-based, and proposed schemes.....	124
Figure 27: Percentage of expired NAL units in different SNR quality layers from 120 random channel realizations of EDF-based, CMSE-based, and proposed schemes.....	126
Figure 28: Percentage of expired NAL units in different temporal layers from 120 random channel realizations of EDF-based, CMSE-based, and proposed schemes.....	127
Figure 29: Average video PSNR of the EDF-based, CMSE-based, and proposed schemes over 120 random channel realizations.....	129
Figure 30: Per-frame video quality comparison between the proposed, EDF-based and CMSE-based schemes for Stefan at pre-roll delay of 0.1s.....	131
Figure 31: Per-frame video quality comparison between the proposed, EDF-based and CMSE-based schemes for Stefan at pre-roll delay of 0.4s.....	132

## LIST OF TABLES

Table 1: Goodness of Fit Statistics for Maximized Likelihood Function .....	34
Table 2: Final Model Factors and Coefficients.....	38
Table 3: Normalized $\overline{CMSE}$ , $\overline{CMSE}_i$ , for Slices in Different Priorities of Sample Videos.....	56
Table 4: Various Combinations of Cross-Layer FEC Coding Schemes.....	63
Table 5: Optimum Cross-Layer Parameters for S-I Scheme, at C = 1.4Mbps.....	71
Table 6: Optimum Cross-Layer Parameter for S-II Scheme, C = 1.4MBPS .....	73
Table 7: Optimum Cross-Layer Parameters for S-III Schemes, C = 1.4Mbps.....	73
Table 8: Optimum Cross-Layer Parameters for S-IV Schemes, C = 1.4Mbps .....	74
Table 9: Optimal Cross-Layer Parameters for S-IV at C = 1.4Mbps for Akiyp Sequence.....	80
Table 10: Bit rates (Kbps) of sub-streams of (a) Table Tennis and (b) Stefan.....	117
Table 11: (a) Average NAL unit sizes, and (b) average CMSE values of Table Tennis.....	118
Table 12: (a) Average NAL unit sizes (bytes) and (b) average CMSE values of Stefan.....	119
Table 13: Average normalized CMSE values of (a) Table Tennis and (b) Stefan.....	123

## SUMMARY

The wireless networks play a critical role in net-centric warfare, including the sharing of the time-sensitive battlefield information among military nodes for situational awareness purpose. However, it is very challenging to organize a low-delay, reliable, infrastructure-less wireless network in the presence of highly dynamic network topology, heterogeneous nodes, intermittent transmission links and dynamic spectrum allocation. The QoS-aware, cross-layer protocols are *key enablers* in effectively deploying the military wireless network.

This report discusses the design of cross-layer protocols for the transmission of delay-sensitive and prioritized data in wireless networks; these protocols consider the QoS issues in an end-to-end fashion and collaboratively design protocols at different network layers. We have used the H.264 compressed video packets as an example of the prioritized and delay-sensitive data.

First, a novel cross-layer scheme is discussed which minimizes the expected received video distortion by jointly optimizing the packet sizes at the application (APP) layer and estimating their forward error correction (FEC) code rates to be allocated at the physical (PHY) layer for bit-rate limited and noisy channels. The optimization considers the source bit rate, packet priority, latency, channel bandwidth and SNR. To reduce the delays, the proposed scheme is also extended to work on each video frame independently by predicting its expected channel bit budget using a generalized linear model. Second, a cross-layer FEC scheme is discussed, which jointly optimizes the Raptor codes at APP layer and rate compatible punctured convolutional (RCPC) codes at PHY layer for the prioritized video packets, in order to minimize the distortion

for the given source bit rates and channel constraints (i.e., SNR and available bandwidth). Our results demonstrate that both these schemes outperform the competing schemes in the literature, and provide significantly better video quality over bit-rate limited and lossy wireless channels.

Finally, a video slice CMSE and deadline-aware sliding-window based scheduling algorithm is designed, which exploits the temporal and SNR scalability of a H.264/SVC compressed bit stream for transmission over a wireless link with time-varying bit rate. This scheme effectively trades off the importance of the network abstraction layer (NAL) units of video bit stream with their deadlines and determines a good transmission order for them. The proposed scheduling scheme reduces the whole frame losses by taking into consideration the relative importance and time-to-expiry of the NAL units, and thereby provides graceful degradation in bad channel conditions.

# 1.0 INTRODUCTION

## 1.1 Motivation

The Air Force (AF) Wireless Networks (also denoted as *military networks* in this report) must be capable of supporting the diverse AF missions, platforms, and communications transport needs of the future. The network can vary from a single airborne node (such as aircraft) connected to a ground station to support voice or low speed data, to a constellation of hundreds of aircrafts and UAVs transporting high speed imagery and real-time collaborative voice and video. The network connections may be point-to-point, broadcast, or multipoint/multicast. The connections could be established either based upon a prearranged network topology, or autonomously without prearrangements, and dynamically as opportunities and needs arise. Key inter-node connectivity functions include the backbone connectivity, subnet connectivity and network access connectivity [1].

The robust multimedia representation and QoS-aware cross-layer network protocols are *key enablers* in effectively deploying the military network infrastructure. The military assets (such as UAVs, surveillance and fighter aircrafts, satellites, ground units) need to (i) share the time-sensitive information (such as battlefield surveillance data/voice/image/video, ally pilots' voice/data, command and control information) among themselves for situational awareness purpose, and (ii) transfer it to the remotely located command and control center. The challenge in military networks is to organize a *low-delay, reliable*, infrastructure-less wireless network in the presence of highly dynamic network topology (due to very high flying speeds), heterogeneous air assets, intermittent transmission links and dynamic spectrum allocation [1].

## **1.2 Objectives**

This report discusses the design of cross-layer protocols for the transmission of delay-sensitive and prioritized data in wireless networks; these protocols consider the QoS issues in an end-to-end fashion and collaboratively design protocols at different network layers. We have used H.264 compressed video packets as an example of the prioritized data. The objectives of this report are:

- i. Use the robust H.264 video bitstream for error-prone wireless channels, including the video packet formation, real-time packet priority assignments, and partial packet decoding.
- ii. Show the importance of real time packet priority assignment for improving QoS in cross-layer protocol design.
- iii. Study the efficacy of a novel cross-layer priority-aware payload adaptation scheme for the prioritized video data.
- iv. Study the performance of a novel cross-layer FEC assignment scheme for prioritized video data.
- v. Study the performance of a novel cross-layer packet scheduling scheme for prioritized video data.

## **1.3 Organization of Report**

Section 1 provides the motivation for this effort. Section 2 introduces the background and assumptions of the techniques presented in this report, including the issues in cross layer design

of wireless network protocols, impact of other layers on these protocols, and need for designing multimedia bitstream.

Our objective in Section 3 is minimizing the expected received video distortion by jointly optimizing the packet sizes at the application (APP) layer and estimating their FEC code rates to be allocated at the physical (PHY) layer for noisy channels. Some low priority slices are also discarded in order to increase the protection to more important slices and meet the channel bit-rate limitations. To avoid the delays associated with optimizing the packet sizes and their associated FEC code rates for entire slices of a GOP, we extend the proposed scheme to work on each frame independently by predicting its expected channel bit budget using a generalized linear model (GLM). The simulation results show that the proposed schemes efficiently transmit the prioritized video over AWGN channels.

The unequal error protection (UEP) has shown promising results for transmitting the prioritized data over error-prone wireless channels. In Section 4, we present a cross-layer design of forward error correction (FEC) schemes by using the UEP Raptor codes at APP layer and UEP rate compatible punctured convolutional (RCPC) codes at PHY layer for the prioritized video packets. A genetic algorithm (GA) based optimization algorithm is proposed to find the optimal parameters for both Raptor and RCPC codes, in order to minimize the video distortion and maximize the peak signal-to-noise-ratio (PSNR) for the given video bit rates and channel constraints (i.e., SNR and available bandwidth). We evaluate the performance of four combinations of the UEP schemes for H.264/AVC encoded video sequences over the AWGN and Rayleigh fading channels and show the superiority of the optimized cross-layer UEP FEC scheme.



In Section 5, we discuss a video slice CMSE and deadline aware sliding-window based scheduling algorithm, which exploits the temporal and SNR scalability of a H.264/SVC compressed bit stream for transmission over a wireless link with time-varying bit rate. The proposed algorithm determines how many and which particular NAL units, from a window of temporal and quality layers, are to be scheduled for transmission during every transmission time interval (TTI). Our algorithm effectively trades off the importance of the NAL units with their deadlines and determines a good transmission order for the NAL units in the sliding window. Our scheduling algorithm reduces the whole frame losses by taking into consideration the relative importance and time-to- expiry (TTE) of the NAL units of different temporal and SNR quality layers, and thereby provides graceful degradation in bad channel conditions.

In Section 6, the conclusions, contributions, future research and recommendation are discussed.

## 2.0 BACKGROUND AND ASSUMPTIONS

H.264/AVC video codec is the most widely used video compression standard jointly developed by the ITU and ISO [2, 3]. However, compressed video transmission is highly vulnerable to packet losses in wireless networks. Lost video packets induce different levels of quality degradation due to temporal and spatial dependencies in the compressed bitstream. An important problem which affects video quality is error propagation where an error in a reference frame propagates to future reconstructed frames which are predicted from that reference frame. This problem has led to the design of error-resiliency features such as flexible macroblock ordering (FMO), data partitioning, and error concealment schemes in H.264 [2, 4, 5].

Though H.264 error-resiliency features reduce the distortion from packet losses, they are still decoupled from various network-centric QoS provisions. QoS support involves several areas, ranging from applications, terminals, and networking architectures to network management, business models, and finally the main target, end users [6]. Enabling QoS in an environment involving mobile hosts under different wireless access technologies is very challenging, because the available resources (e.g., bandwidth, battery life, etc.) in wireless networks are scarce and dynamically change over time. Since the capacity of the channel in a wireless network varies randomly with time, providing deterministic QoS for video is not only difficult but will also likely result in conservative guarantees and waste of resources. Hence, statistical QoS guarantees in terms of received video quality, goodput based on successfully received data, probability of packet loss, and packet delay have gained importance. There are

several fundamental challenges in supporting the end-to-end QoS for video delivery over wireless networks [6–8]:

1. QoS support depends on a wide range of technological aspects, including video coding, high-performance physical and link layer support, efficient packet delivery, congestion control, error control, and power control.

2. Different applications have diverse QoS requirements in terms of data rates, delay bounds, and packet loss probabilities. For example, unlike non-real-time data packets, video services are sensitive to packet delivery delay but can tolerate some transmission errors and even frame losses.

3. Different types of networks have different characteristics, usually referred to as network heterogeneity. The network conditions, such as bandwidth, packet loss ratio, delay, and delay jitter, vary over time in a wireless environment. Bit-error rate (BER) in a wireless network is much higher than in the wireline network. Moreover, link layer error control schemes, such as automatic repeat request (ARQ), are widely used to overcome wireless channel errors; this further increases the dramatic variation of bandwidth and delay in wireless networks. To make things even more complicated, the packet loss in wireless networks can be caused by either congestion leading to buffer overflow or by a noisy channel leading to packet errors.

4. There is dramatic heterogeneity among end users in terms of latency requirements, visual quality, processing capabilities, power, and bandwidth. It is thus a challenge to design a delivery mechanism that not only achieves efficient resource utilization but also meets the heterogeneous requirements of the end users.

To address the above challenges, the QoS requirement should be supported in all components of the video delivery system using a cross-layer perspective, which include (a) QoS provisioning from networks, (b) scalable and/or prioritized video presentation from applications, and (c) network adaptive congestion/error/power control. To deliver the best end-to-end performance for such wireless systems, video coding, reliable transport and wireless resource allocation must be considered jointly, thus moving from the traditional layered system architecture to a cross-layer design. Broadly, this report addresses cross-layer QoS issues for video packet delivery over wireless links through: (1) prioritized transmission control schemes that can derive and adjust the bit-budget for prioritized video data, and (2) cross-layer QoS adaptation that can optimally choose statistical QoS guarantees for each video priority class of a prioritized transmission system so as to provide better video quality. Adaptation of packet size and forward error correction (FEC) are two well-known techniques to combat packet loss due to channel impairments. In this report, we use them as QoS adaptation techniques for prioritized video data. Packet size adaptation can be carried out at different layers such as APP, transport, and medium access control (MAC) layers. FEC adaptation can be carried out at the APP and PHY layers.

Packet size adaptation calls for a trade-off between reducing the total number of overhead bits by using large packets and reducing the transmission error rate by using small packets. However, maximum throughput does not guarantee the minimum video distortion at the receiver due to the following reason - Unlike data packets, loss of H.264 compressed video packets induces different amounts of distortion in the received video. Therefore the packet size should be adaptive to the packet priority. However, existing payload (i.e., packet size) adaptation schemes

in the literature do not consider the distortion contribution of the packet. Packet size adaptation can be carried out at the APP layer by aggregating the smaller-sized network abstraction layer (NAL) units belonging to the different priority classes into packets of different sizes. However, there is an upper bound on the size of the APP layer packets known as maximum transmission unit (MTU) size for wireless networks.

Recent research has demonstrated the promise of cross-layer protocols for supporting the QoS demands of multimedia applications over wireless networks [9-11]. Van der Schaar et al. [10] discuss different cross layer solutions and extend the MAC-centric approach to demonstrate that the joint APP-MAC-PHY approach is best suited for transmitting multimedia (e.g., video streaming) over wireless networks. The joint APP-MAC-PHY cross-layer interface is desirable to achieve our objective of QoS adaptation by using the channel noise information, bit rate constraints, and network packet size limitation.

## **2.1 Modeling the Impact of other Layers on Cross-Layer Protocols**

The protocols must consider the close interaction among different layers, beginning with PHY as discussed below:

- *Application-level QoS parameters* such as source data rates, latency (real-time vs. non-real-time), loss sensitivity, constant bit-rate vs. variable bit-rate. For this one should consider the characteristics of compressed H.264 AVC video bitstreams in terms of their scalability (frame-rate, frame-size, fine granularity scalability), error resiliency (data partitioning, resynchronization, interleaving, etc.), packetization, metadata, packet scope, packet priority, etc. [4, 12-14].

- *Network-level QoS parameters* such as available bandwidth, link BER and packet loss rates, flow priority [12-13]. Please note that the values of these parameters will considerably vary due to the spectrum mobility and dynamic topologies.
- *Effect of PHY* including the spectrum sensing delays and spectrum mobility. Each channel could suffer from varying interference levels and noise. The modulation (BPSK, QPSK, etc.) and code rates (1/2, 1/3, etc.) also depend on channel conditions and required QoS. Another important aspect is the channel heterogeneity as different channels may be located on widely separated slices of spectrum with different bandwidths and different propagation characteristics [15-18].
- *Effect of data link layer*: presence of common channel signaling, scheduling, channel access delays, connection establishment and management policies to adapt to spectrum mobility and sharing. Similarly, the choice of CDMA vs. OFDM and the effect of Doppler on multiplexing schemes [18].

Since there are too many parameters, many of them inter-dependent, a small set of metrics could be used to consider the cost of a configuration for the protocol layer. For example, one possibility is to measure the cost of configurations as some weighted combination of data rate, transmission delay, error rates, etc.

## **2.2 Design of Cross-Layer Rate Control, Payload Adaptation, Packet Scheduling and FEC Protocols**

The QoS-aware Rate Control, Payload Adaptation, Packet Scheduling, and FEC schemes are essential for reliable video transmission over wireless networks. However, the existing schemes

do not simultaneously consider the characteristics of video bitstreams (such as packet priority, choice of scalability, etc.), network (such as congestion and collision), PHY (such as channel error rates, available bandwidth, choice of hierarchical modulation) and the end-user QoS requirements in a cross-layer fashion. As a consequence, these schemes fail to provide the end-to-end rate control for reliable transmission of prioritized packets whose loss would cause significant fluctuations in the video signal quality.

Video priority-aware schemes based on the video bitstream, network and PHY characteristics are likely to provide better performance. Selective packet rescheduling/retransmission could be applied for high priority packets. The encoder can use more powerful FEC schemes (i.e., rate of the channel codes is adapted according to the packet priority) or switch to a different frequency or channel. As a result, the FEC codes rates and fragmentation sizes should be jointly optimized for prioritized video bitstream and the effect of NALU size should be studied on the received video quality for various channel losses. The network simulation tool (ns-2) can be used to simulate a multi-user and multi-hop wireless ad hoc network. Performance metrics of interest include the received video quality (PSNR and VQM) for a specified bit-rate, buffer size as well as the channel and congestion-induced packet losses.

## **3.0 CROSS-LAYER PRIORITY-ADAPTIVE PACKETIZATION AND ERROR CORRECTION FOR WIRELESS CHANNELS**

### **3.1 Introduction**

Adapting the packet size to channel error characteristics improves the successful packet transmission probability and reduces retransmissions [19-21]. It involves a trade-off between reducing the number of overhead bits by using large packet sizes and reducing the transmission error rate by using small packet sizes. Maximizing throughput in this manner does not guarantee minimum received video distortion since lost video packets can induce significantly different amounts of distortion. Hence, video packet size should also be adaptive to the packet importance. However, existing payload (i.e., packet size) adaptation schemes in the literature do not consider distortion contribution of the packet [22].

In this section, we describe our cross-layer scheme which minimizes the expected received video distortion by jointly optimizing the packet sizes at the APP layer and estimating their FEC code rates to be allocated at the PHY layer for noisy channels. Some low priority slices are also discarded in order to increase the protection to more important slices and meet the channel bit-rate limitations. Our proposed scheme ensures that higher priority slices which contribute more distortion are sent in smaller packets with stronger FEC coding. At the same time, it also efficiently controls the overhead incurred from the total protocol header bits associated with the formed packets. The distortion contributed by each slice is determined by its CMSE. Simulation results show that the proposed scheme efficiently transmits video over noisy channels.



To avoid the delays associated with optimizing the packet sizes and their associated FEC code rates for entire slices of a GOP, we extend our scheme to work on each frame independently by predicting its expected channel bit budget. This prediction uses a GLM developed over the factors (a) normalized CMSE per frame, (b) channel SNR, and (c) normalized compressed frame bit budget allocated by the H.264 encoder. The three factors are determined from a video dataset that spans high, medium, and low motion complexity. Further, to avoid the complexity associated with computing the CMSE distortion contributed by a video slice, we use our low-complexity GLM defined in [23] for predicting the slice CMSE.

### **3.1.1 Contributions**

Existing schemes do not consider different distortion contributions (e.g., CMSE-driven priority) of video slices while computing their packet size and FEC code rate, nor do they discard low priority slices. Our scheme has the following distinguishing features: (i) minimizes the video distortion by jointly optimizing the packet size and FEC code rate for a given source video bit rate, channel bit rate and channel SNR; (ii) adapts packet size and FEC code rate to the distortion contribution (i.e., CMSE-driven priority) of video slices; (iii) discards some low priority slices to improve protection to high priority slices and meet the channel constraints; and (iv) performs real-time optimization over slices of each frame by using the predicted slice CMSE and frame overhead bit budget values for live streaming applications.

## **3.2 Related Work**

Packet headers and protocol layer overhead reduce the effective throughput. The need for adapting the payload length and data rate is discussed in [34]. To address the variation in network conditions, solutions for adaptive packet size adjustments at the APP layer have been

discussed in [19 - 21, 24 - 29]. The effect of packet size on the loss rate and delay characteristics in a wireless real-time application was studied in [20]. It was shown that APP level packet size optimization could facilitate efficient usage of wireless network resources, improving the service provided to all end users sharing the network.

Choi et al. [24] designed cross-layer schemes to study the effect of optimal packet size, MAC layer retransmissions, and APP layer FEC on multimedia delivery over wireless networks. They noted that the packet size is tightly related to the packet delay and channel conditions. An algorithm that allows an ARQ protocol to dynamically optimize the packet size based on the wireless channel bit error rates was proposed in [19]. Lee et al. [21, 25] developed an analytic model to evaluate the impact of channel BER on the quality of streaming a MPEG-4 video with fine granular scalability. They proposed a video transmission scheme, which combines the adaptive assignment of packet size with unequal error protection (UEP) to increase the end-to-end video quality.

Shih [26, 29] proposed a scheme which integrated the packet size control mechanism with the optimal packet-level FEC in order to enhance the efficiency of FEC over wireless networks. Both the degree of FEC redundancy and the transport packet size were adjusted simultaneously in accordance with a minimum bandwidth consumption strategy to transmit video frames with delay bound and target frame error rate constraint. Lin et al. [27] formulated an optimization problem to minimize the required resource units for a single user by adjusting payload length, modulation, block size, and code rate for wireless channels. An adaptive packet and block length FEC control mechanism is discussed in [28]. Lin and Cosman [30] studied code rate allocation

with slice discarding for pre-encoded H.264 video slices of a group of pictures (GOP). Each slice consisted of a horizontal row of macroblocks and was considered to be an independent packet.

In [34], authors presented a mathematical framework to maximize a single user throughput by using the symbol rate, the packet length, and the constellation size of the modulation. In [31, 32], authors provided a theoretical framework without retransmission to optimize single user throughput by adjusting the source bit rate and payload length as a function of channel conditions. However, the maximal throughput transmission does not ensure the packet error rate (PER) requirement. A cross-layer design considering retransmission was discussed in [46]. Authors optimized the length of payload and suggested the associated physical transmission modes, which include modulation and coding scheme, for a given channel SNR.

### 3.3 Methods, Assumption and Procedures

#### 3.3.1 Proposed Cross-Layer Approach

Figure 1 illustrates a flow diagram of our proposed cross-layer approach at the transmitter. The APP layer carries out two functions: CMSE based slice prioritization and optimal packet formation (illustrated further in Figure 2) for H.264 video slices.

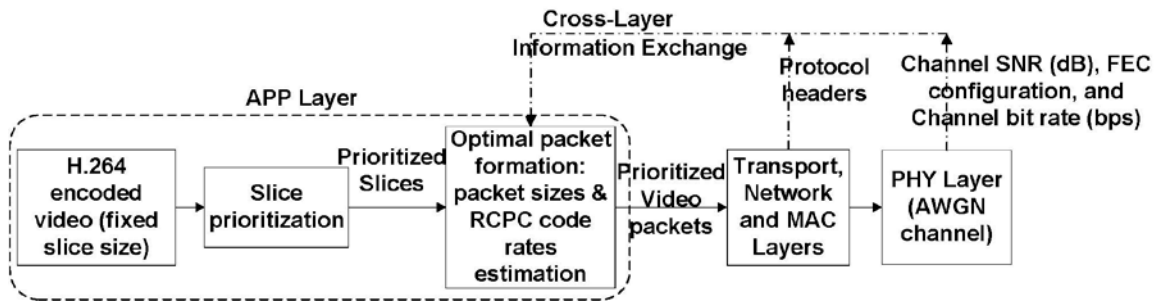


Figure 1: Flow diagram of proposed cross-layer system.

### 3.3.1.1 CMSE Computation/Prediction of H.264 Video Slices

The video frames in a GOP are encoded using the fixed slice size configuration in H.264/AVC, where MBs of a frame are aggregated into slices with fixed size [2]. The loss of a slice in a reference frame can introduce error propagation in the current and subsequent frames until the end of GOP. We compute the total distortion introduced by the loss of a slice by using the cumulative mean squared error (CMSE), which takes into consideration the error propagation within the entire GOP. Let the original uncompressed video frame at time  $t$  be  $f(t)$ , and the decoded frame without and with the slice loss be  $\widehat{f(t)}$  and  $\widetilde{f(t)}$ , respectively. Assuming that each slice consists of  $M$  macroblocks consisting of  $16 \times 16$  pixels, the MSE introduced by the loss of a slice is given by

$$\frac{1}{256M} \sum_{m=1}^M \sum_{i=1}^{16} \sum_{j=1}^{16} [(pixel\_value_{m,i,j})_{\widehat{f(t)}} - (pixel\_value_{m,i,j})_{\widetilde{f(t)}}]^2. \quad (1)$$

Here,  $(m, i, j)$  represents the pixel at coordinate  $(i, j)$  for the  $m$ th macroblock. The CMSE contributed by the loss of the slice is computed as the sum of MSE over the current and all the subsequent frames in the GOP. However, the computation of slice CMSE introduces high computational overhead as it requires decoding the entire GOP for every slice loss. This overhead can be avoided by predicting the slice CMSE using our low-complexity GLM recently proposed in [23]. This model reliably predicts the slice CMSE values by extracting the encoded frame and the error frame features. The encoded frame features consist of motion characteristics,

signal characteristics, maximum residual energy, and total number of MB sub-partitions in a slice. The error frame features consist of the temporal duration, initial mean square error, and initial structural similarity index. The actual slice CMSE values were used as ground truth. The readers are encouraged to refer to [23] for more details. The slice contributing the highest distortion is the most important slice (i.e., highest priority). This process defines the relative importance order for the slices in the GOP. Note that our joint video packetization and error protection scheme proposed in this section will also work well with other slice distortion computation schemes such as Li and Liu [43].

### **3.3.1.2 H.264 Video Packet Formation**

The optimal packet formation block uses a joint optimization scheme to form variable-sized packets (by aggregating pre-encoded slices according to their CMSE) and estimate their corresponding optimal FEC code rates that are applied at the PHY layer, in order to minimize the received video distortion as will be discussed in Section 3.3.2.

The FEC configuration contains a mother code rate and a family of rate compatible punctured convolutional (RCPC) code rates [39]. We use binary phase shift keying (BPSK) modulation and the packet size is constrained by the wireless network MTU [52]. The optimal packet formation block uses the information about the MTU size, RTP/UDP, IP and MAC layer headers which remain unchanged for a given network, and the channel SNR, FEC configuration and channel bit rate information from the PHY layer. The RTP/UDP/IP overhead appended to each packet formed at the APP layer is four bytes after robust header compression (RoHC) [51]. Each packet is also appended with 50 bytes of MAC and PHY layer headers. Our scheme studies

the video quality improvement that can be achieved by exploiting the slice priorities and the trade-offs between the priority-adaptive packet sizes and RCPC code rates with the total incurred overhead (FEC + network protocol header) for a given channel SNR, channel bit rate, and source bit rate.

### 3.3.2 Expected Video Distortion Minimization

We introduce a DP-based approach to minimize the expected video distortion.  $R_{CH}$  is the channel transmission rate in bits per second. The video is encoded at a frame rate of  $f_s$  fps. The total outgoing bit budget for a GOP of length  $L_G$  frames is  $\frac{R_{CH}L_G}{f_s}$ . We use  $n_s$  to denote the total number of slices generated within a GOP;  $n_s$  is a constant. We use  $n_p$  to denote the number of packets formed from these slices in the GOP;  $n_p$  is variable.  $S_p(i)$  is the  $i^{th}$  packet size before adding network headers of size  $h$  bits and parity bits from the selected RCPC code. The RCPC code rates are chosen from a candidate set,  $\mathbf{R}$ , of punctured code rates  $\{R_1, R_2, R_3, \dots, R_K\}$ . The number of packets discarded is  $n_{pd}$  which will be described in the following sections.

### 3.3.2.1 Packet Formation (PF) Block

The proposed scheme, denoted as **DP-UEP**, is a recursive process between two blocks: Packet formation (PF) block and Optimal RCPC code rate allocation (OCRA) block as shown in Figure 2. The PF block initializes  $n_p = n_s$  and  $n_{pd} = 0$ , and calls the OCRA block after sorting the  $n_p = n_s$  packets of a GOP in descending priority order. The OCRA block determines the optimal RCPC packet code rates and the number of packets discarded,  $n_{pd}$ , to minimize a dual cost function value (computed over the GOP) described in Section 3.3.2.2. The OCRA block then forwards the computed parameters to the PF block as shown in Figure 2.

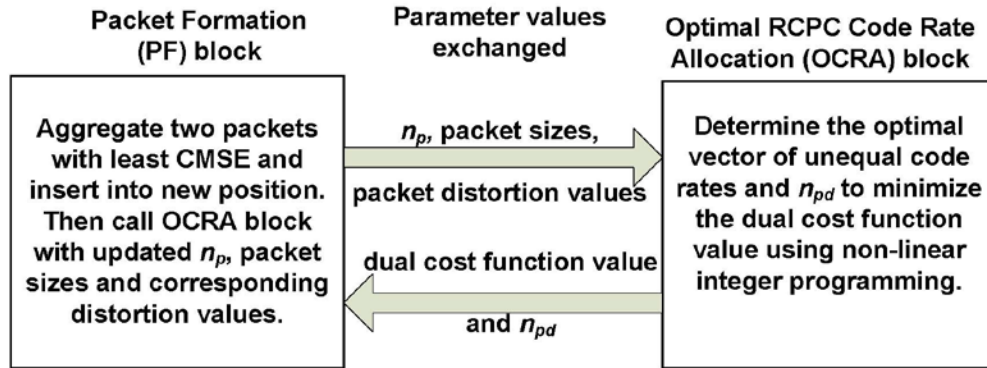


Figure 2: Block diagram of proposed dynamic programming approach.

The PF block aggregates the two packets with least CMSE contribution from the remaining set of packets not discarded by the OCRA block. The aggregated packet is inserted into a new position in the sorted list based on its distortion computed as the sum of the CMSE values of both packets. This maintains the decreasing order of packet distortion. It calls the

OCRA block again to determine optimal RCPC code rates for the new set of packets. The parameters shown in Figure 2 are exchanged recursively between the blocks until aggregating packets is no longer beneficial to reduce the dual cost function value. As an example, Figure 3 shows one iteration of our proposed scheme in the PF block. The first packet in each iteration is the most important and contributes the maximum distortion. After returning from the OCRA block, the number of packets is updated to  $n_p = n_s - n_{pd}$  since  $n_{pd}$  packets were dropped in the OCRA block. The two least important packets are then aggregated and inserted into a new position while the remaining packets are simply retained. The aggregated packet is at position  $n_p - j$ . The  $n_p - 1$  packets with their sizes and distortion values are once again sent to the OCRA block, to estimate their new optimal packet code rates.

The size of the aggregated packets is constrained by the MTU size for wireless networks. Aggregating packets reduces the total overhead from network protocol headers; the bits saved are used to increase the FEC protection to more important packets. Since the PF block aggregates the least important packets, this ensures that packets contributing higher distortion are transmitted with smaller sizes, and the OCRA block ensures that they have stronger FEC hence lower packet error probabilities.



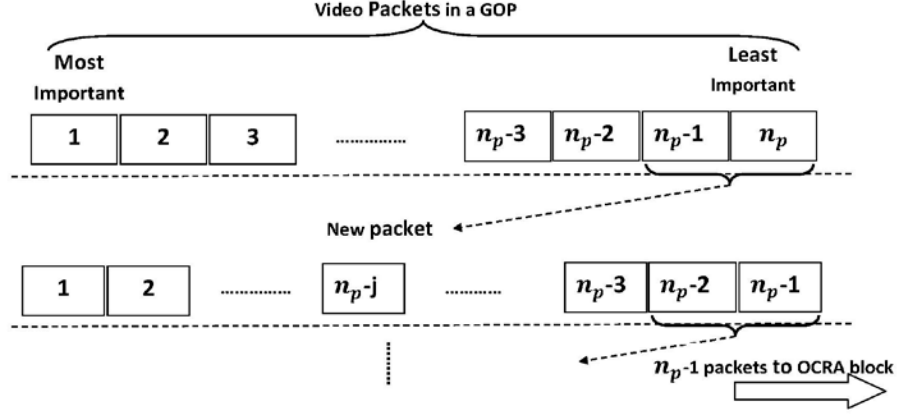


Figure 3: Packet formation in PF block.

### 3.3.2.2 Distortion Minimization with OCRA Block

The distortion due to the compression is neglected in this formulation because the slices are pre-encoded and assumed to be at relatively high quality, so compression distortion is small compared to distortion from slice losses and discards. The initial values are  $n_p = n_s$  and  $n_{pd} = 0$ . The expected video distortion within a GOP,  $E[\tilde{D}_{GOP}]$ , is modeled as the sum of the distortion due to channel-induced packet loss and distortion from packets discarded at the sender as in [30].

$$E[\tilde{D}_{GOP}] = \sum_{i=1}^{n_p - n_{pd}} E[\tilde{D}_p(i)] + \sum_{i=n_p - n_{pd} + 1}^{n_p} D_p(i) \quad (2)$$

$D_p(i)$  is the distortion caused due to the loss of packet  $i$  and is computed as the sum of the CMSE of individual slices contained in the packet. Each video packet is appended with a  $h$  bit network header and parity bits for a code rate  $r_i$  selected from the set  $\mathbf{R}$ . We consider a discrete-

time memoryless AWGN channel. A video packet is in error if at least one bit is in error after channel decoding at the receiver. If the bit errors following decoding were independent from bit to bit, then the packet error probability,  $p_{pkt}(i)$ , which depends on the channel SNR, packet size, and the selected RCPC code rate could be computed as in [30, 32, 34, 38, 40]:

$$p_{pkt}(i) = 1 - (1 - p_b(SNR, r_i))^{\left(\frac{h+S_p(i)}{r_i}\right)} \quad (3)$$

where  $p_b(SNR, r_i)$  is the bit error probability after channel decoding for code rate  $r_i$ . We use the above expression for packet error probability in the design procedure to determine the FEC rates.

For a given value of  $n_{pd}$ , the distortion due to the discarded packets in Equation (2) is a constant

$K_1$ . The optimization problem for minimizing expected video distortion over the GOP by

allocating optimal code rates is formulated as:

$$\begin{aligned} & \min_{\mathbf{r}} \left\{ \sum_{i=1}^{n_p - n_{pd}} \left[ 1 - (1 - p_b(SNR, r_i))^{\left(\frac{h+S_p(i)}{r_i}\right)} \right] D_p(i) + K_1 \right\} \\ & = K_1 + \min_{\mathbf{r}} \left\{ \sum_{i=1}^{n_p - n_{pd}} \left[ 1 - (1 - p_b(SNR, r_i))^{\left(\frac{h+S_p(i)}{r_i}\right)} \right] D_p(i) \right\} \\ & \quad \text{subject to} \\ & \quad (C1) \quad \sum_{i=1}^{n_p - n_{pd}} \frac{h + S_p(i)}{r_i} \leq \left( \frac{R_{CHLG}}{f_s} \right) \\ & \quad (C2) \quad r_{i-1} \leq r_i \quad \text{for } i = 2, 3, 4, 5, 6, \dots, (n_p - n_{pd}) \\ & \quad \text{where } \mathbf{r} = [r_1, r_2, \dots, r_{n_p - n_{pd}}] \text{ and } r_i \in \mathbf{R} \end{aligned} \quad (4)$$

Constraint 1 in Equation (4) is the channel bit rate constraint. Constraint 2 ensures that higher priority packets have code rates at least as good as those allocated to lower priority packets. This speeds up the optimization process by narrowing down the selection set of packet code rates. To solve this non-linear integer programming problem, we first relax the constrained optimization problem in Equation (4) to an unconstrained problem [37, 42]. By absorbing the constraints into the objective using Lagrange multipliers

$\lambda = [\lambda_1, \lambda_2, \dots, \lambda_{n_p - n_{pd}}]$  with each  $\lambda_i \in \mathbb{R}^+$ , we construct the Lagrangian cost function as:

$$\begin{aligned}
 F_{GOP}(\mathbf{r}, \lambda) = & K_1 + \sum_{i=1}^{n_p - n_{pd}} \left[ 1 - (1 - p_b(SNR, r_i))^{\left(\frac{h + S_p(i)}{r_i}\right)} \right] D_p(i) \\
 & + \lambda_1 \left( \sum_{i=1}^{n_p - n_{pd}} \frac{h + S_p(i)}{r_i} - \frac{R_{CHLG}}{f_s} \right) + \sum_{i=2}^{n_p - n_{pd}} \lambda_i (r_{i-1} - r_i) \\
 \text{where } \lambda = & [\lambda_1, \lambda_2, \dots, \lambda_{n_p - n_{pd}}]
 \end{aligned} \tag{5}$$

We form the dual cost function  $d_{GOP}(\lambda)$  by minimizing the Lagrangian cost function for a given  $\lambda$ , where  $\lambda$  is searched using a subgradient approach which will be discussed in the following section. Let  $\mathcal{C}$  be the space of all possible combinations of  $r_i, i = 1, 2, \dots, n_p - n_{pd}$  selected from  $\mathbf{R}$  that can be applied to the packets before transmission. The dual function is computed as:

$$\begin{aligned}
d_{GOP}(\lambda) &= \min_{\mathbf{r} \in \mathcal{C}} F_{GOP}(\mathbf{r}, \lambda) \\
&= \min_{\mathbf{r} \in \mathcal{C}} \sum_{i=1}^{n_p - n_{pd}} \left\{ \left( 1 - (1 - p_b(SNR, r_i))^{\left(\frac{h+S_p(i)}{r_i}\right)} \right) D_p(i) \right\} \\
&\quad + \lambda_1 \left( \sum_{i=1}^{n_p - n_{pd}} \frac{h+S_p(i)}{r_i} - \frac{R_{CHLG}}{f_s} \right) + \sum_{i=2}^{n_p - n_{pd}} \lambda_i (r_{i-1} - r_i) + K_1 \\
&= K_2 + \min_{\mathbf{r} \in \mathcal{C}} \sum_{i=1}^{n_p - n_{pd}} \left( \left( 1 - (1 - p_b(SNR, r_i))^{\left(\frac{h+S_p(i)}{r_i}\right)} \right) D_p(i) + \lambda_1 \left( \frac{h+S_p(i)}{r_i} \right) \right) \\
&\quad + \sum_{i=2}^{n_p - n_{pd}} \lambda_i (r_{i-1} - r_i)
\end{aligned} \tag{6}$$

$K_2 = K_1 - \lambda_1 \left( \frac{R_{CHLG}}{f_s} \right)$  in Equation (6) is a constant and the computation of  $d_{GOP}(\lambda)$  can be

further simplified as follows.

Let  $A(r_i) = D_p(i) \left( 1 - (1 - p_b(SNR, r_i))^{\left(\frac{h+S_p(i)}{r_i}\right)} \right) + \lambda_1 \left( \frac{h+S_p(i)}{r_i} \right)$ . Then we can modify the first

term in Equation (6) as:

$$\begin{aligned}
&\min_{\mathbf{r} \in \mathcal{C}} \left\{ \sum_{i=1}^{n_p - n_{pd}} A(r_i) + \sum_{i=2}^{n_p - n_{pd}} \lambda_i (r_{i-1} - r_i) \right\} \\
&= \min_{\mathbf{r} \in \mathcal{C}} A(r_1) + A(r_2) + \dots + A(r_{n_p - n_{pd}}) + \lambda_2 (r_1 - r_2) + \lambda_3 (r_2 - r_3) + \\
&\quad \dots + \lambda_{n_p - n_{pd}} (r_{n_p - n_{pd} - 1} - r_{n_p - n_{pd}}) \\
&= \min_{r_1 \in \mathbf{R}} \{A(r_1) + \lambda_2 (r_1)\} + \min_{r_2 \in \mathbf{R}} \{A(r_2) - \lambda_2 (r_2) + \lambda_3 (r_2)\} \\
&\quad + \dots + \min_{r_{n_p - n_{pd} - 1} \in \mathbf{R}} \left\{ A(r_{n_p - n_{pd} - 1}) - \lambda_{n_p - n_{pd} - 1} (r_{n_p - n_{pd} - 1}) + \lambda_{n_p - n_{pd}} (r_{n_p - n_{pd} - 1}) \right\} \\
&\quad + \min_{r_{n_p - n_{pd}} \in \mathbf{R}} \left\{ A(r_{n_p - n_{pd}}) - \lambda_{n_p - n_{pd}} (r_{n_p - n_{pd}}) \right\} \\
&= \min_{r_1 \in \mathbf{R}} \{A(r_1) + \lambda_2 (r_1)\} + \sum_{i=2}^{n_p - n_{pd} - 1} \min_{r_i \in \mathbf{R}} \{A(r_i) + r_i (\lambda_{i+1} - \lambda_i)\} \\
&\quad + \min_{r_{n_p - n_{pd}} \in \mathbf{R}} \left\{ A(r_{n_p - n_{pd}}) - \lambda_{n_p - n_{pd}} (r_{n_p - n_{pd}}) \right\}
\end{aligned}$$

The dual function can now be expressed in terms of function  $A(r_i)$  as:

$$\begin{aligned}
d_{GOP}(\lambda) &= K_2 + \min_{r_1 \in \mathbf{R}} \{A(r_1) + \lambda_2(r_1)\} + \sum_{i=2}^{n_p - n_{pd} - 1} \min_{r_i \in \mathbf{R}} \{A(r_i) + r_i(\lambda_{i+1} - \lambda_i)\} \\
&\quad + \min_{r_{n_p - n_{pd}} \in \mathbf{R}} \{A(r_{n_p - n_{pd}}) - \lambda_{n_p - n_{pd}}(r_{n_p - n_{pd}})\} \\
&= K_2 + \sum_{i=1}^{n_p - n_{pd}} \min_{r_i \in \mathbf{R}} \tilde{F}_{GOP,i}(r_i, \lambda_i)
\end{aligned} \tag{7}$$

$$\text{where } \tilde{F}_{GOP,i}(r_i, \lambda_i) = \begin{cases} A(r_1) + \lambda_2(r_1) & \text{for } i = 1 \\ A(r_i) + r_i(\lambda_{i+1} - \lambda_i) & \text{for } i = 2, 3, 4, \dots, n_p - n_{pd} - 1 \\ A(r_{n_p - n_{pd}}) - \lambda_{n_p - n_{pd}}(r_{n_p - n_{pd}}) & \text{for } i = n_p - n_{pd} \end{cases}$$

The minimum of the dual cost function for a given  $\lambda$  can be found by minimizing the sub-Lagrangian cost functions  $\tilde{F}_{GOP,i}(r_i, \lambda_i)$  individually. The solution space of the minimization of  $F_{GOP}(\mathbf{r}, \lambda)$  is  $(K + 1)^{(n_p - n_{pd})}$ . Since we can minimize the sub-Lagrangians individually,  $d_{GOP}(\lambda)$  can be computed with only  $(n_p - n_{pd})(K + 1)$  evaluations of  $\tilde{F}_{GOP,i}(r_i, \lambda_i)$  and comparisons [42]. This reduces the computational complexity involved in deriving the optimal set of packet sizes and their code rates. The frame-based optimization schemes use the slices of a frame (instead of a GOP) to form  $n_p$  packets. Therefore, their optimization complexity is much smaller than for a GOP-based scheme.

### 3.3.2.3 Determination of $\lambda$

We use the subgradient method [42] to search for the best  $\lambda$  over the space  $\mathcal{C}$ . The dual function  $d_{GOP}(\lambda)$  is a concave function of  $\lambda$  even when the problem in the primal domain is not convex [37, 42]. Therefore the optimal  $\lambda$  is found by solving  $\max_{\lambda \in \mathbb{R}^+} d_{GOP}(\lambda)$ . Since the dual is a piecewise linear concave function, it may not be differentiable at all points. Nevertheless, subgradients can still be found and are used to compute the optimal value. It can be shown that the subgradient is a descent direction of the Euclidean distance to the set of maximum points of the dual function [42]. This property is used in the subgradient method for the optimization of a non-smooth function. The subgradient method is an iterative search algorithm for  $\lambda$ . In each iteration,  $\lambda_i^{k+1}$  is updated by the subgradient  $\xi_i^k$  of  $d_{GOP}(\lambda)$  at  $\lambda_i^k$ :

$$\lambda_i^{(k+1)} = \max(0, \lambda_i^k + s_k \xi_i^k / \|\xi^k\|) \quad (8)$$

where  $s_k$  is step size. Based on the derivation in [42], the subgradients  $\xi^k$  of  $d_{GOP}(\lambda)$  at  $\lambda^k$  are

$$\begin{aligned} \xi_1^k &= g(\mathbf{r}^k) - \frac{R_{CHLG}}{f_s} = \sum_{i=1}^{n_p - n_{pd}} \left( \frac{h + S_p(i)}{r_i} \right) - \frac{R_{CHLG}}{f_s} \\ \xi_i^k &= r_{i-1} - r_i \text{ for } i = 2, 3, 4, \dots, n_p - n_{pd} \end{aligned} \quad (9)$$

where  $g(\cdot)$  is the rate constraint function of the problem and  $\mathbf{r}^k = [r_1^k, r_2^k, \dots, r_{n_p - n_{pd}}^k]$  is the solution to the term  $\min_{\mathbf{r} \in \mathcal{C}} F_{GOP}(\mathbf{r}, \lambda^k)$  in Equation (6).

### 3.3.2.4 Discarding Packets

By explicitly discarding a small number of low priority packets, we gain additional room for packet size adaptation and FEC, and can derive significant benefits overall. To allow either the discarding of less important packets or sending them unprotected, the candidate set of punctured code rates  $\mathbf{R}$  is modified to  $\{1, R_1, R_2, R_3, \dots, R_K, \infty\}$ . This neither changes the objective function to be minimized in Equation (4) nor does it affect the optimization algorithm discussed in Section 3.3.2.2. If the code rate of packet  $i$ ,  $r_i = \infty$ , then its probability of bit error  $p_b(SNR, r_i) = 1$  causing it to be discarded. The induced distortion is accounted for in the overall expected distortion  $E[\tilde{D}_{GOP}]$  through component  $K_1$  in Equation (4). If  $r_i = 1$ , the video packet is transmitted uncoded.

### 3.3.3 Frame-Level DP-UEP Scheme

The **DP-UEP** scheme discussed in Section 3.3.2 was designed for a pre-encoded video and the cross-layer optimization was performed over each GOP. Its computational complexity and delay are not suitable for live streaming applications, such as live sports events. In this section, we extend **DP-UEP** to be applied over the slices of a single frame instead of the entire GOP to reduce its computational complexity and delay. This requires **DP-UEP** to process the encoded slices of only one frame at a time in the PF and OCRA blocks (shown in Figure 2) instead of performing optimization over the slices of an entire GOP. Since a typical GOP consists of different frame types (i.e., IDR, I, P, and B), we require a good estimate of the

channel bit budget for that frame in order to allocate the protocol header and FEC bits to its packets. Moreover, different frame types generate different numbers of slices that contribute different amounts of distortion based on the error propagation and video content. Therefore, we need to distribute the channel bit budget for a GOP among the different frames and to that extent we study the video factors which are most influential on the expected channel bit budget estimate of a frame. From now on, we refer to our **DP-UEP** scheme over the slices of the entire GOP as **DP-UEP(GOP)** and over slices of a frame as **DP-UEP(frame)**.

Before investigating the important factors influencing the expected channel bit budget for each frame within the GOP, we study how well **DP-UEP(frame)** might perform compared to **DP-UEP(GOP)**. We study the average PSNR and average VQM performance of **DP-UEP(frame)** by using the measured slice CMSE values and the channel bit budget allocated to each frame by the **DP-UEP(GOP)** for Foreman and Silent. Later in Section 3.3.4, we train a GLM for predicting the expected channel bit budget for each frame in real-time. To avoid the delays involved with processing an entire GOP, we will need to use an estimate of the frame bit budget rather than the actual bit budget allocated by the **DP-UEP(GOP)** scheme. However, analyzing the channel bit budget allocation,  $R_l$  for the frame  $l$ , by the **DP-UEP(GOP)** scheme can provide some motivation for whether the frame-based approach is worth pursuing. To compute  $R_l$ , we first derive the overhead bit budget proportion,  $w_{ovh}^l$  for the frame  $l$ , from the result of the **DP-UEP(GOP)** scheme as:

$$w_{ovh}^l = \frac{\text{\#FECbitsforframe } l}{\text{\#FECbitsforwholeGOP}} \quad (10)$$



This quantity, while it is explicitly the fraction of FEC bits which a particular frame gets relative to FEC bits for the whole GOP, is taken to be an estimate of overhead bits (both FEC and protocol header bits) which the frame gets relative to the overhead bits for the whole GOP.

$R_l$  is then evaluated using  $w_{ovh}^l$  for a video bit rate denoted by  $R_v$  as:

$$R_l = \sum_{i=1}^{n_s^l} S_p^l(i) + w_{ovh}^l \left\{ \frac{(R_{CH} - R_v)L_G}{f_s} \right\} \quad (11)$$

where  $n_s^l$  is the number of slices in frame  $l$  and  $S_p^l(i)$  is the size of slice  $i$  in frame  $l$ . The video bit rate of 720 Kbps, used in our simulations in Section 3.4, is assigned to  $R_v$ . We determine the optimal packet sizes and their corresponding code rates separately for each frame in the GOP using the cross-layer DP-based approach. We observe that the average PSNR performance of **DP-UEP(frame)** is only slightly lower than that of **DP-UEP(GOP)** (shown later in Figure 8), but still higher than the **Dual15** scheme. A small drop in average PSNR and VQM is due to the fact that our optimization scheme for slices of each frame is sub-optimal compared to the DP-UEP(GOP) scheme. In other words, DP-UEP(frame) may have discarded some slices from a frame which were retained in the DP-UEP(GOP) scheme.

From the analysis of the **DP-UEP(GOP)** scheme, we observed that  $w_{ovh}^l$  for a frame  $l$  is dependent on the following video factors: (a) normalized CMSE for frame  $l$ , denoted as  $w_{cmse}^l$ , (b) normalized compressed frame bit budget, denoted as  $w_c^l$ , (c) channel SNR, and (d) video

content.  $w_{cmse}^l$  is computed as a ratio of the total CMSE contribution of all slices in frame  $l$  to the total CMSE contribution of all slices in the GOP.  $w_c^l$  is computed as the ratio of the size of the compressed frame in bits to the total source bit rate for the GOP.

$$w_{cmse}^l = \frac{\sum_{i=1}^{n_s^l} D_p^l(i)}{\sum_{j=1}^{L_G} \sum_{i=1}^{n_s^j} D_p^j(i)} ; w_c^l = \frac{\sum_{i=1}^{n_s^l} S_p^l(i)}{\left(\frac{R_{vLG}}{f_s}\right)} \quad (12)$$

where  $D_p^l(i)$  is the distortion caused due to the loss of slice  $i$  in frame  $l$ .

### 3.3.4 Frame-Level DP-UEP using Prediction

The **DP-UEP**(frame) scheme in the previous section has the following two major issues for live streaming applications: (i) measuring CMSE values of the slices of a frame requires the decoding of current and other frames of the GOP which is computationally intensive and introduces about one GOP time delay, and (ii) determining the channel bit budget for different frames in each GOP in real-time. In this section, we introduce an improved frame-level scheme, denoted as **DP-UEP**(predict), to address these issues.

CMSE Prediction: For the first issue, we use a slice CMSE prediction scheme proposed in [23], which predicts the CMSE corresponding to individual slice losses of a frame in real-time. This scheme uses a combination of video parameters which can be easily extracted during the encoding of a frame without requiring information from future frames.

$\hat{w}_{ovh}^l$  Prediction: To address the second issue we train a GLM to predict the  $w_{ovh}^l$  of every frame  $l$ , denoted as  $\hat{w}_{ovh}^l$ , in real-time. The GLM to estimate  $\hat{w}_{ovh}^l$  is developed over a database of the factors discussed in Section 3.3.3 and derived for videos with different types of motion and content. We use a database of 12 CIF video sequences that span (a) low motion: *Silent*, *Mother-Daughter*, *Bridge*, and *Akiyo*; (b) medium motion: *Table Tennis*, *Coastguard*, *Tempete*, and *Foreman*; and (c) high motion: *Soccer*, *Bus*, *Football*, and *Stefan*. We use the first three sequences from each motion category for training and the last one from each category for testing. For a given  $R_v$ , we compute the factors  $w_{ovh}^l$ ,  $w_{cmse}^l$ , and  $w_c^l$  for the frames of each training video sequence by using the **DP-UEP**(GOP) scheme and store them in the database along with the channel SNR. The GLM, explained in the following section, is trained offline only once.  $\hat{w}_{ovh}^l$  is then used to estimate the channel bit budget constraint (as shown in Equation (11)) and estimate the optimal packet sizes and code rates for the slices of frame  $l$ .

### 3.3.4.1 GLM Approach for Estimating $w_{ovh}^l$

GLMs are an extension of classical linear models [41, 45]. We train the GLM to predict  $w_{ovh}^l$  (i.e.,  $\hat{w}_{ovh}^l$ ). Let  $\mathbf{Y} = [y_1, y_2, y_3, \dots, y_N]$  be a vector of our response variable  $w_{ovh}^l$  from the database. Every data point  $y_i$  in  $\mathbf{Y}$  is expressed as a linear combination of a known covariate

vector  $[1, x_{i1}, x_{i2}, x_{i3}, \dots, x_{ip}]$ , where  $p$  is the number of factors, and a vector of unknown regression coefficients  $\boldsymbol{\beta} = [\gamma, \beta_1, \beta_2, \dots, \beta_p]^T$ . The covariate vector is a row of matrix  $\mathbf{X}$  of order  $N \times (p + 1)$  with elements  $x_{ij}$  for  $N$  observations and  $p$  factors also from the database.

$$f(\mathbf{Y}) = \mathbf{X}\boldsymbol{\beta} \quad ; \quad f(y_i) = \gamma + \sum_{j=1}^p x_{ij}\beta_j. \quad (13)$$

where  $f(\cdot)$  is called the link function. After estimating  $\boldsymbol{\beta}$ , we use it to derive the predicted response variable vector  $\hat{\mathbf{Y}} = [\hat{y}_1, \hat{y}_2, \hat{y}_3, \dots, \hat{y}_N]$  computed as  $f^{-1}(\mathbf{X}\boldsymbol{\beta})$ ;  $f^{-1}$  is the inverse of the link function and  $\hat{\mathbf{Y}}$  is a vector of  $\hat{w}_{ovh}^l$ .

### 3.3.4.2 Response Variable Distribution

To determine the link function for the GLM, we need to know the distribution family of our response variable. We evaluate the goodness of fit for ranking Weibull, Gamma, and Gaussian fitted distributions of  $w_{ovh}^l$  by using three information criteria (IC): (a) SIC: Schwarz information criterion, aka Bayesian information criterion [47], (b) AIC: Akaike information criterion [35, 36], and (c) HQIC: Hannan-Quinn information criterion [40]. Each information criterion depends on the number of distribution parameters to be estimated. For example, the Gaussian distribution has two parameters, mean and standard deviation, and the Gamma and Weibull distributions have two parameters, scale and shape parameter. Each information

criterion also depends on the number of observations of our response variable  $w_{ovh}^l$ , and the maximized log-likelihood estimate of the fitted distribution producing the set of observations. For  $m$  observations and  $u$  distribution parameters, the SIC is the most strict in penalizing loss of degrees of freedom by having more distribution parameters and is computed as  $u \times \ln(m) - 2 \times \ln(L_{max})$ , where  $L_{max}$  is the maximized value of the likelihood function for the fitted distribution. HQIC holds the middle ground in its penalizing for  $u$  and is computed as  $\left(\frac{m-2u+2}{m-u+1}\right) \ln(\ln(m)) - 2 \times u \times \ln(L_{max})$ . Finally, AIC is the least strict of the three in penalizing loss of degrees of freedom and is computed as  $\left(\frac{m-2u+2}{m-u+1}\right) - 2 \times \ln(L_{max})$ .

Table 1: Goodness of Fit Statistics for Maximized Likelihood Function

IC/Fitted Distribution	Weibull	Gamma	Gaussian
SIC	23.71	23.71	25.79
HQIC	12.72	12.72	16.88
AIC	6.25	6.25	8.33

We randomly chose  $m = 5000$  observations from the vector of  $w_{ovh}^l$  values in the database, obtained from all the training videos at channel SNRs from -2 dB to 6 dB. These are divided into 100 bins from zero to one and the likelihood function is maximized for each of the three fitted distributions. The distribution parameters where the likelihood is maximized are: (a)

Gaussian: mean = **0.05**, standard deviation = **0.095**, (b) Gamma: shape parameter = **1**, scale parameter = **0.05**, and (c) Weibull: shape parameter = **1**, scale parameter = **0.05**. Since the shape parameter of both Gamma and Weibull distributions is 1, they are in essence exponential distributions. In Table 1, the goodness of fit of all three information criteria are minimum for Weibull and Gamma distributions; therefore our response variable is exponential. Figure 4 also shows that the cumulative distributions of Weibull and Gamma are the same and closer to the cumulative distribution of the 5000 observations than the Gaussian cumulative distribution.

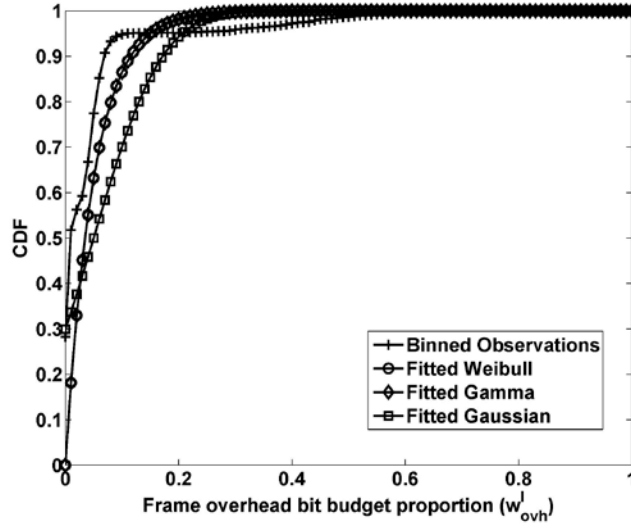


Figure 4: Cumulative distribution function (CDF) for the binned observations and fitted distributions of  $w_{ovh}^l$ .

### 3.3.4.3 Model Fitting and Validation

We use the statistical software R [53] for fitting our GLM and its validation. We classified our response variable as a member of the exponential family of distributions with

identity as its link function. The GLM model in R uses the AIC index to determine the order in which three factors,  $w_{cmse}^l$ ,  $w_c^l$ , and channel SNR are fitted. Here, the AIC index is defined as  $2p - 2\max(L)$ , where  $p$  is the number of factors and  $L$  is the log-likelihood estimate for the model. We let  $\mathbf{Y}^k$  represent the model with a subset of  $k$  factors. The  $i^{th}$  data point in  $\mathbf{Y}^k$ ,  $y_i^k$ , where  $i = 1, 2, \dots, N$  is expressed as:

$$y_i^k = \gamma + \beta_1^k x_{i1} + \beta_2^k x_{i2} + \dots + \beta_j^k x_{ij} + \dots + \beta_k^k x_{ik}. \quad (14)$$

Here,  $\gamma$  is the intercept as considered in Equation (13),  $\beta_j^k = 1, 2, \dots, k$  are the fitted coefficients for  $k$  factors, and  $x_{ij}$  represents the  $j^{th}$  factor value for the  $i^{th}$  observation in  $\mathbf{Y}^k$ .

The simplest model is the Null Model having only the intercept  $\gamma$  whereas the Full Model has all the  $p$  factors, i.e.  $k = p$ . The factors are also known as covariates. The following forward stepwise approach is used to determine the order of our covariates:

Step 1: We fit a group of  $p$  univariate models and compute their AIC values. The best univariate model has the smallest AIC value.

Step 2: We then fit  $(p - 1)$  multivariate models where each model has two covariates. The first covariate is from the best univariate model in Step 1 and the second covariate is chosen from the remaining  $(p - 1)$  available covariates. We compute the AIC values for the  $(p - 1)$  multivariate models and choose the best multivariate model with the smallest AIC value. The two covariates fitted at this stage would progress to the next step to be fitted with the third covariate.

The covariates and coefficients of our final model are shown in Table 2. We also introduced two interactions,  $w_c^l \times \text{channel SNR}$  and  $w_{cmse}^l \times \text{channel SNR}$ .

The goodness of fit for a GLM can be characterized by its deviance, which is a general term of variance [45]. By definition, the deviance is zero for the Full model and positive for all other models. A smaller deviance means a better model fit. After fitting a particular model, the importance of each factor in the model can be evaluated by the resultant increase in deviance when we remove that factor from the model. The third column in Table 2 shows the reduction in deviance as each of the covariates in the first column is added to the model using the stepwise approach described above. Model 1 is the best univariate model with  $w_c^l$ . Model 2 has both  $w_c^l$  and  $w_{cmse}^l$  covariates. In addition to these, Model 3 has channel SNR. Model 4 adds the first interaction between  $w_c^l$  and channel SNR, and Model 5 includes all the factors in Table 2.



Table 2: Final Model Factors and Coefficients

Covariate(Factor)	Coeff. for Final Model	Model Deviance
$\gamma$	<b><math>-0.0193</math></b>	167.18
$w_c^l$	1.3240	12.2
$w_{cmse}^l$	<b><math>5.37 \times 10^{-5}</math></b>	11.7
channel SNR	<b>0.0028</b>	11.7
$w_c^l \times$ channel SNR	-0.0564	9.75
$w_{cmse}^l \times$ channel SNR	<b><math>9.521 \times 10^{-5}</math></b>	9.7

### 3.3.5 Problem Formulation of other Error Protection Schemes

We compare our proposed DP-UEP schemes discussed in Sections 3.3.2 and 3.3.3, with the **Dual15** [30], and the **EEP-slice-ENH** schemes. The **Dual15** scheme treats every slice as a packet and does not aggregate them to save on the total overhead incurred from network protocol headers of 54 bytes being associated with every slice. It finds the optimal set of punctured code rates to protect the slices based on their importance (i.e., using UEP) and minimize expected received video distortion.

The **EEP-slice-ENH** is similar to our proposed scheme **DP-UEP** in the way pre-encoded slices are aggregated to form packets with more important ones having smaller sizes and error probabilities and also the less important packets being discarded to meet the channel bit rate constraint. However, unlike **DP-UEP**, all packets in **EEP-slice-ENH** are equally protected with

the best possible EEP code rate. This scheme is broadly similar to other packet (or payload) size adaptation schemes in the literature [19, 20, 24, 27, 50]. The objective of this scheme is to minimize the expected received video distortion and it is formulated in a manner similar to Equation (4):

$$\begin{aligned}
& \min_{\mathbf{r} \in \mathbf{R}} \left\{ \sum_{i=1}^{n_p - n_{pd}} \left[ 1 - (1 - p_b(\text{SNR}, \mathbf{r}))^{\left(\frac{h+S_p(i)}{r}\right)} \right] D_p(i) + K_1 \right\} \\
& = K_1 + \min_{\mathbf{r} \in \mathbf{R}} \left\{ \sum_{i=1}^{n_p - n_{pd}} \left[ 1 - (1 - p_b(\text{SNR}, \mathbf{r}))^{\left(\frac{h+S_p(i)}{r}\right)} \right] D_p(i) \right\} \\
& \quad \text{subject to} \quad \sum_{i=1}^{n_p - n_{pd}} \frac{h+S_p(i)}{r} \leq \left( \frac{R_{CHLG}}{f_s} \right)
\end{aligned} \tag{15}$$

Constraint 2 in Equation (4) is not valid here since  $\mathbf{r}$  is no longer a vector. As in Equation (4),  $K_1$  is the permanent distortion caused by the discarded packets and is constant for a given value of  $n_{pd}$ . Apart from the change that only a single  $\lambda$  and  $\mathbf{r}$  value needs to be determined, the same DP-based approach described in the previous sections is used to solve the optimization problem in Equation (15).

### 3.4 Results and Discussion

In this section, we evaluate and compare the performance of our proposed **DP-UEP** schemes with **Dual15**, and **EEP-slice-ENH** schemes with video quality measured by PSNR and VQM [46, 49].

### 3.4.1 Simulation Setup

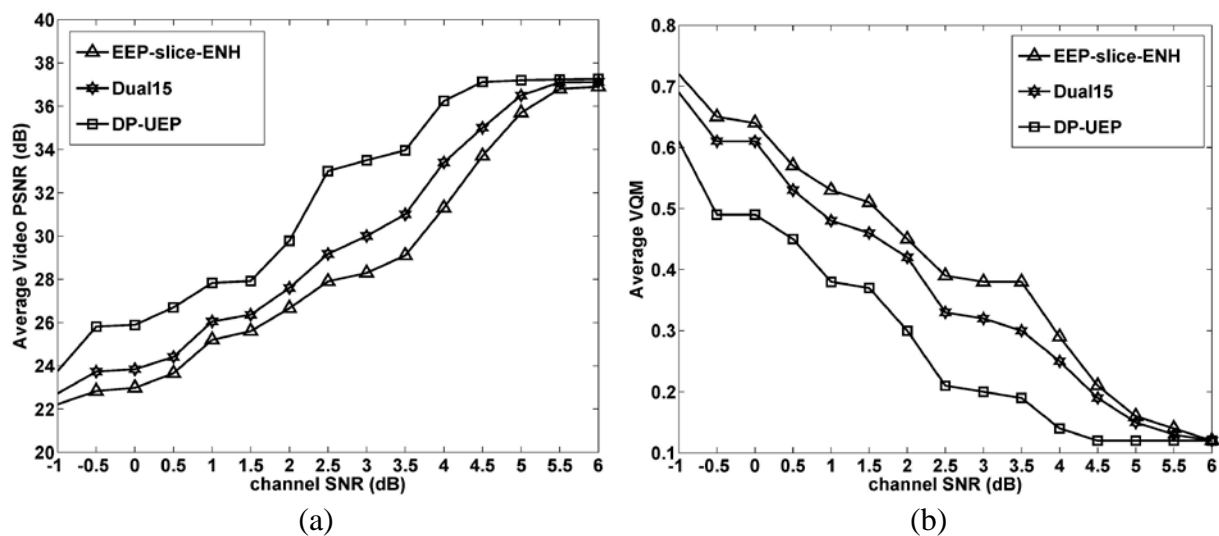
Two CIF (352 x 288) video sequences, Foreman and Silent, are used in our experiments. Silent has lower motion activity than Foreman. They are encoded using H.264/AVC JM 18.5 reference software for a GOP length of 20 frames with GOP structure IDR B P B ... P B IDR at 30 frames/sec (fps), at an encoding rate of 720 Kbps and transmitted over a 2 Mbps AWGN channel. The slice size in the fixed slice size configuration of H.264/AVC is set to 300 bytes and the slices are formed using dispersed mode FMO with two slice groups. Two reference frames are used for predicting the P and B frames, with error concealment enabled using temporal concealment and spatial interpolation. The error concealment in a frame depends on the frame type and the type of losses encountered. If an entire frame (IDR, P or B) is lost, first the motion vectors and reference indices of the co-located MBs in the previously decoded reference frame are copied and then motion compensation is used to reconstruct the lost frame based on the copied motion information. If some slices of a predicted (P or B) frame are lost, the decoder verifies the availability of motion vector information for the lost MBs. If the motion vectors are available, motion copy is performed else co-located MBs of the previous reference frame are directly copied. If some slices of an IDR frame are lost, the corresponding MBs are concealed using spatial interpolation. Error concealment is enabled for all the schemes evaluated in this section.

The total network protocol header size is 54 bytes per packet. The mother code of the RCPC code has rate  $1/4$  with memory  $M=4$  and puncturing period  $P=8$ . Log-likelihood ratio (LLR) is used in the Viterbi decoder. The initial RCPC rates available are  $\{(8/9), (8/10), (8/12), (8/14), (8/16), (8/18), (8/20), (8/22), (8/24), (8/26), (8/28), (8/30), (8/32)\}$ . Two additional rates,

8/8 corresponding to no coding and  $\infty$  corresponding to discarding are also included. The performance evaluation of the schemes is based on a bit-level simulation of the compressed videos using the derived packet sizes and FEC code rates over 100 realizations of every AWGN channel SNR. The simulation results use the CMSE values computed from Equation 1.

### 3.4.2 Performance of DP-UEP Scheme

Figure 5 shows the average PSNR and VQM performance over an AWGN channel. As the channel SNR increases, the packet error decreases and the received videos achieve average PSNRs closer to their error-free PSNR values. The **EEP-slice-ENH** scheme performs the worst. Though it adapts the packet size to the video priority by aggregating the slices and discarding lower priority packets, it is still limited to providing equal protection to all the packets formed. The lowest and highest optimal EEP code rates derived across GOPs were (8/20, 8/14). However, as the channel SNR deteriorates in Figure 5, the lowest code rate 8/20 is insufficient to protect the packets from channel induced errors.



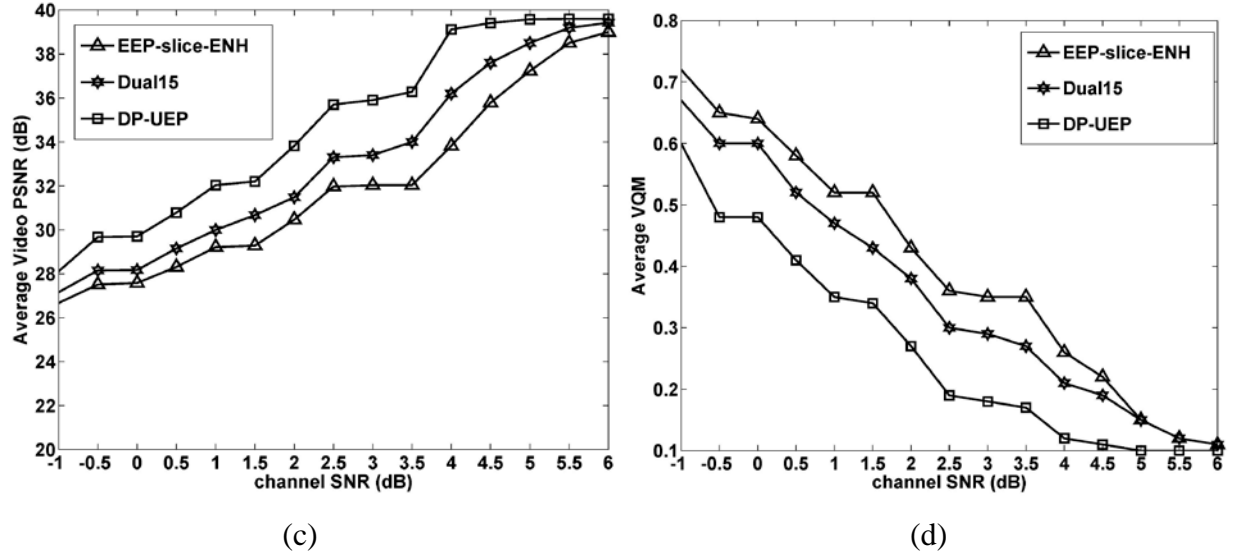


Figure 5: Average video PSNR (dB) and corresponding average VQM comparison computed over 100 realizations of each AWGN channel for Foreman:(a),(b), and Silent:(c),(d).

The **Dual15** scheme does not consider packet formation through slice aggregation and only performs optimal (UEP) RCPC code rate allocation to the slices (considered as individual packets) of each GOP [30]. It also discards least important slices, if required to meet the channel bit budget constraints. The slice error probability in the **Dual15** scheme is dependent on the optimal RCPC code rate allocated since the size of each slice is more or less the same. Also every slice in the **Dual15** scheme is attached with the 54 byte network protocol header resulting in more overhead. In contrast, our proposed **DP-UEP** scheme takes advantage of both the priority-adaptive packet sizes and optimal RCPC packet code rate allocation. Our **DP-UEP** scheme assigns optimal code rates as low as 8/32 to the high priority packets with small packet sizes (e.g. 300 byte, which is the slice size used in encoding, or 600 byte obtained by aggregating two slices) and higher code rates to the lower priority packets with larger packet sizes within every GOP. The packet sizes of the low priority packets are restricted by the network MTU size

of 1500 bytes. Figure 5 shows the improvement in video quality of our **DP-UEP** scheme compared to the **EEP-slice-ENH** and **Dual15** schemes. For example, at a channel SNR of 3 dB, the **EEP-slice-ENH**, **Dual15**, and **DP-UEP** schemes achieve average VQM values of 0.38, 0.32, and 0.2, and corresponding average PSNR values of 28.3 dB, 30.2 dB, and 33.5 dB, for Foreman. Our **DP-UEP** scheme achieves maximum PSNR gains of 3.5 dB for Foreman and 2.8 dB for Silent over **Dual15** at a channel SNR of 3 dB. The **DP-UEP** scheme also achieves maximum gains of 5.2 dB for Foreman and 4.3 dB for Silent over the **EEP-slice-ENH** scheme at channel SNR of 3 dB. Similar behavior is also observed in the VQM performance.

This considerable improvement in video quality achieved by our **DP-UEP** scheme can be explained by the following two factors: (i) the lower number of slices discarded per GOP shown in Figure 6, and (ii) the composition of the final transmitted bits in terms of the compressed source bits, network protocol headers, and FEC bits shown in Figure 7. Balancing the overhead due to the FEC parity bits allows the **Dual15** scheme to discard fewer slices per GOP as compared to the **EEP-slice-ENH** scheme. Our **DP-UEP** scheme further reduces the number of discarded slices as compared to the **Dual15** scheme by balancing both the overhead due to FEC parity bits as well as the network protocol headers attached to the packets formed by aggregating slices. For example, at a channel SNR of 3 dB in Figure 6, our **DP-UEP** scheme does not discard any slices whereas 20 and 35 slices are discarded in every GOP by the **Dual15** and **EEP-slice-ENH** schemes, respectively. As the channel SNR decreases, more slices are discarded by every scheme. For example, at a channel SNR of -1 dB, 101, 62, and 50 slices are discarded by the **EEP-slice-ENH**, **Dual15**, and **DP-UEP** schemes, respectively. This means that though we encode the video at a target bit rate of 720 Kbps, every scheme adjusts this bit rate by discarding

the slices in order to minimize the expected received video distortion under the given channel SNR condition and bit budget constraints.

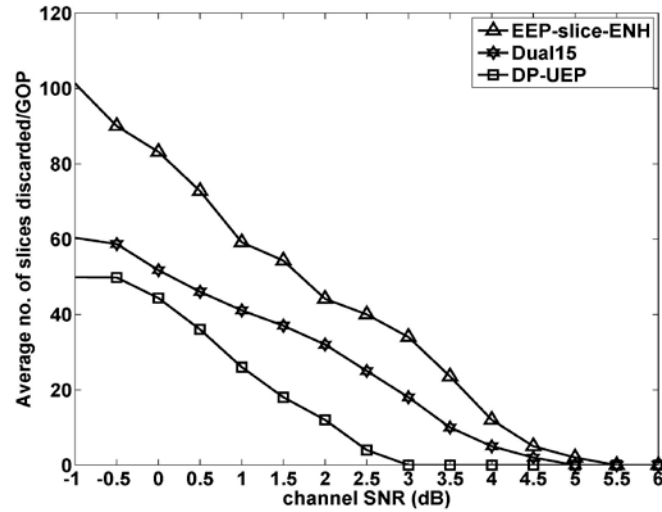


Figure 6: Average number of slices discarded per GOP in EEP-slice-ENH, Dual15, and DP-UEP for Foreman.

Figure 7 shows the bit contribution of the source, network protocol headers, and FEC to the total bits transmitted over a 2 Mbps channel at 3 dB channel SNR for Foreman. Our **DP-UEP** scheme transmits more source bits (i.e., a relatively higher bit rate) than the other two schemes by reducing the network protocol overhead as well as allocating optimal RCPC code rates based on packet priority. It also uses only **5.5%** bits for the network protocol overhead, compared to **8.5%** and **11.5%** overhead bits for EEP-slice-ENH and Dual15, respectively. Further, **61.3%** bits are allocated for FEC overhead by DP-UEP compared to **57.3%** in Dual15, thus providing better FEC protection. Although EEP-slice-ENH uses **64.1%** FEC bits, it uses EEP which

ignores packet priority. The DP-UEP scheme sends the highest percentage of source bits (i.e., 33.2%) which also correlates to no slices being discarded at 3 dB channel SNR, shown earlier in Figure 6. A similar trend is also observed for Silent, and for other channel SNRs.

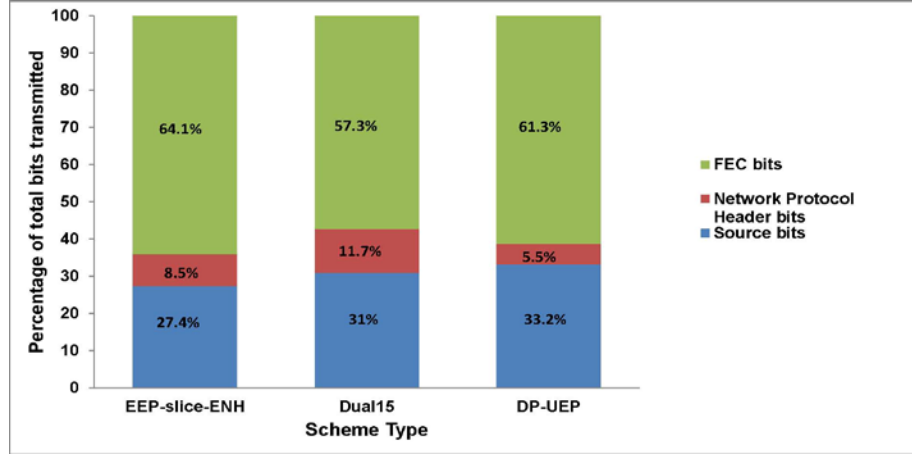


Figure 7: Distribution of the final output bits for Foreman at 3 dB channel SNR in EEP-slice-ENH, Dual15, and DP-UEP schemes.

### 3.4.3 Performance of DP-UEP(frame) Scheme

We evaluate the average PSNR and average VQM of our proposed **DP-UEP(predict)** scheme for the three test videos: low motion Akiyo, medium motion Foreman, and high motion Stefan. The predicted channel bit budget for frame  $l$  is evaluated as  $\hat{w}_{ovh}^l \times \frac{R_{CHLG}}{f_s}$ . The proposed

**DP-UEP(GOP)** scheme in Section 3.3.2 was used to compute the optimal packet sizes and RCPC code rates for the slices of frame  $l$ .  $\hat{w}_{ovh}^l$  uses the coefficients of the factors shown in

Table 2. Since computing the factor  $w_{cmse}^l$  for frame  $l$  is not feasible in real-time,  $\hat{w}_{cmse}^l$  uses the



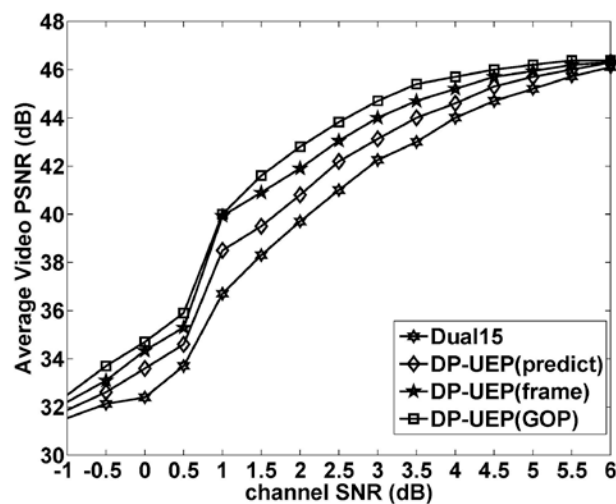
predicted CMSE value of each slice  $i$ ,  $\hat{D}_p^l(i)$  in frame  $l$ , as computed in [23]. But the predicted slice CMSE values of the future frames in the GOP will not be available during real-time transmission. We therefore use the total predicted CMSE of all the slices of the previous GOP to compute the normalized predicted CMSE of the frame in the current GOP as shown in Equation (16) below.

$$\hat{w}_{cmse}^l = \frac{\sum_{i=1}^{n_s^l} \hat{D}_p^l(i)}{\sum_{j=1}^{L_G} \sum_{i=1, \text{previous GOP}}^{n_s^j} \hat{D}_p^j(i)} \quad (10)$$

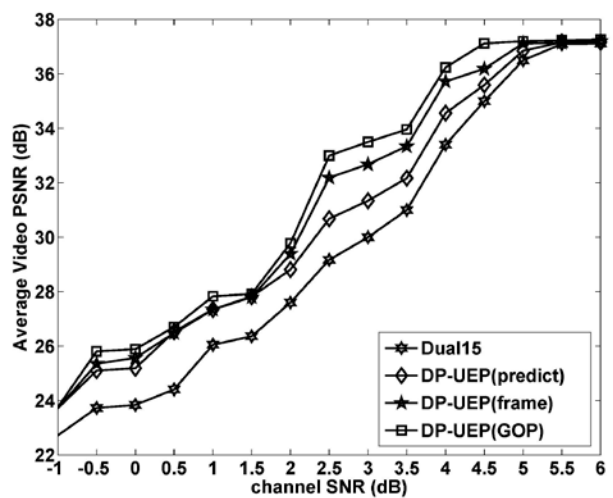
For the first GOP, the  $\hat{w}_{cmse}^l$  is assumed to be zero. It is reasonable to use the predicted CMSE of the previous GOP because for most GOPs there is a high correlation between the CMSE of adjacent GOPs. On a core 2 Duo 2.6 GHz Intel processor with 4GB RAM, we observed that the average computation time across all test videos and channel SNR from -1 dB to 6 dB, is 75 ms for the IDR frame, 10.5 ms for the P frame, and 1.5 ms for the B frame. Since IDR frames have considerably more slices than P and B frames, and P frames have more slices than B frames, the computation time also varies accordingly. These computational delays are acceptable in live streaming applications.

Figure 8 shows the performance of the **DP-UEP(predict)**, **DP-UEP(frame)**, **DP-UEP(GOP)**, and **Dual15** schemes on the test videos, in terms of average PSNR and VQM values. The GOP structure, frame rate, and slice size are the same as considered in Section 3.4.1, and error concealment is also enabled. The videos are encoded at 720 Kbps and transmitted over

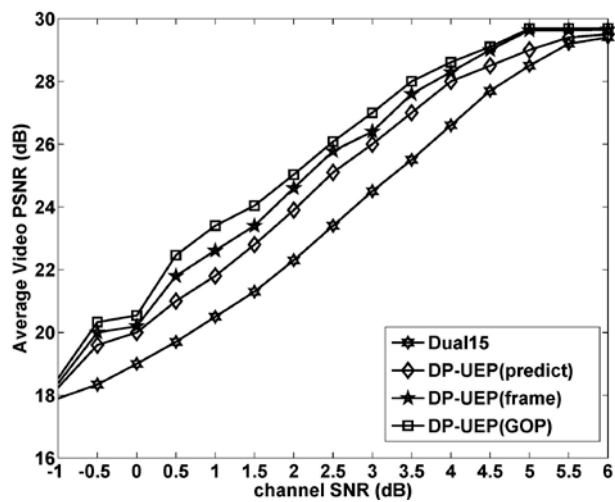
a 2 Mbps AWGN channel. We observe that the error-free PSNR value decreases as the motion in the video increases.



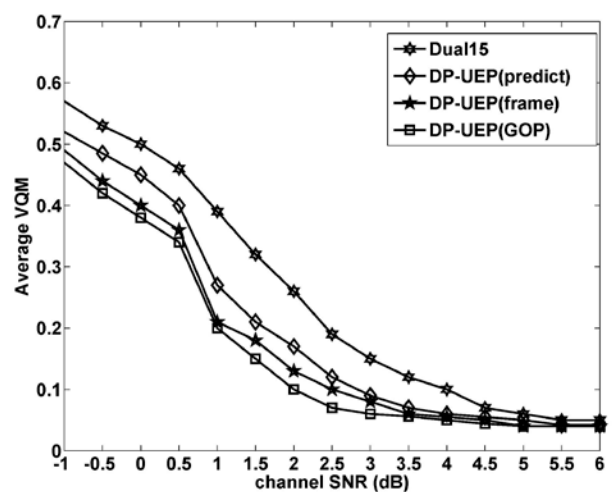
(a)



(b)



(c)



(d)

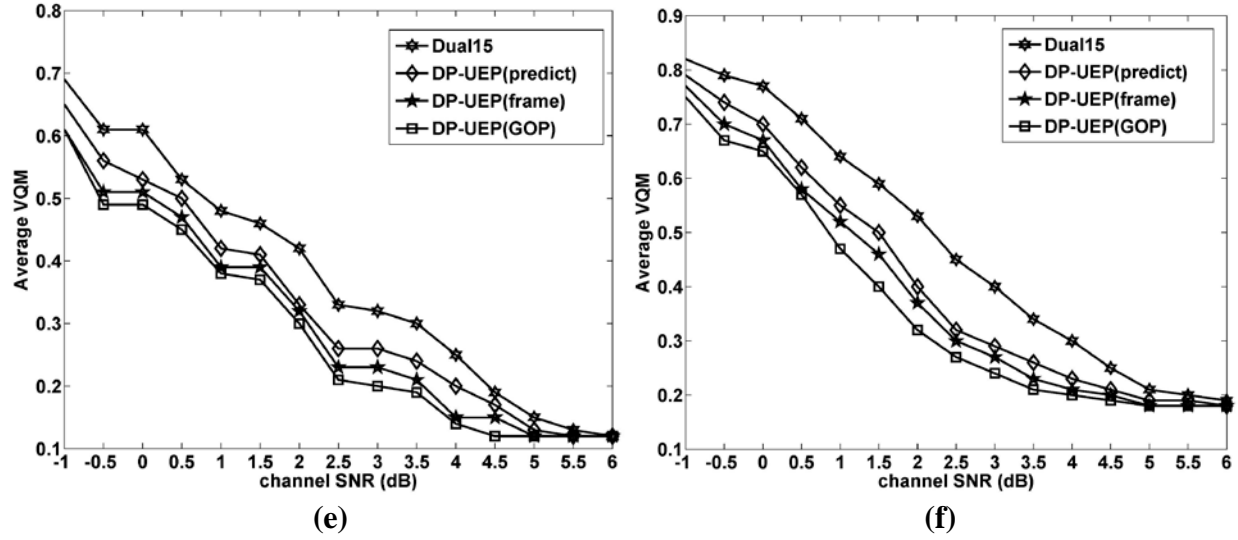


Figure 8: Average video PSNR (dB) and average VQM comparison computed over 100 realizations of each AWGN channel for Akiyo: (a),(d), Foreman: (b),(e) and Stefan: (c), (f). The error-free PSNR values are: 46.5 dB for Akiyo, 37.3 for Foreman, and 29.7 for Stefan.

**DP-UEP(predict)** has better performance than the **Dual15** scheme for all three test videos. **DP-UEP(predict)** enables real-time packet formation and transmission of videos which is not possible with the other three schemes. However, its performance is lower than **DP-UEP(GOP)** and **DP-UEP(frame)** due to the prediction of channel bit budget and slice CMSE values for each frame. For example, the PSNR gain achieved by **DP-UEP(GOP)** over **Dual15** for Foreman in Figure 8 is 3.5 dB at a channel SNR of 3 dB. For **DP-UEP(frame)** which knows the required channel bit budget, the PSNR gain drops to 2.7 dB. Predicting the channel bit budget for each frame in **DP-UEP(predict)** causes the PSNR gain to drop further to 1.4 dB. Similar behavior can also be seen for Akiyo and Stefan in Figures 8. The maximum PSNR gains achieved by **DP-UEP(predict)** over **Dual15** are 1.8 dB for Akiyo at 0.5 dB channel SNR, 2.12 dB for Foreman at 1 dB channel SNR, and 1.5 dB for Stefan at channel SNR of 2.5 dB. Similar

trends are also observed in the VQM performance of the three videos shown in Figure 8. Further, simulations of three more test videos (whale show, Hall Monitor, and Container) from outside our database showed trends similar to those in Figure 8.

### 3.5 Conclusion

An efficient joint optimization algorithm for packet formation and optimal RCPC code rate allocation was proposed to improve the quality of H.264/AVC bitstreams transmitted over noisy channels. The proposed algorithm used a cross-layer information exchange between the PHY, MAC and APP layers. A dynamic programming approach was used where packets were formed through slice aggregation and the optimal RCPC packet code rates were determined recursively over a GOP. The options of not coding or discarding some less important packets were exploited to reduce the expected received video distortion by increasing protection to more important packets. The proposed scheme outperformed EEP schemes as well as our previous scheme in [30], providing significantly better video quality for different sequences. The dynamic programming approach was extended to work on each frame instead of the entire GOP in order to enable live streaming with low computational complexity. The frame bit budget prediction used a GLM model developed using three factors - normalized compressed frame bit budget, normalized frame CMSE and channel SNR over a database of videos. Our proposed dynamic programming approach showed reasonable gains in PSNR and VQM in videos spanning low, medium and high motion. Our proposed schemes can work well with current wireless network standards such as IEEE 802.11n with MTU packet size restrictions. It would be interesting to

evaluate the proposed schemes along with adaptive modulation and coding for time-varying link conditions and channel bit rates.

## 4.0 CROSS-LAYER FEC SCHEME FOR PRIORITIZED VIDEO TRANSMISSION OVER WIRELESS CHANNELS

### 4.1 Introduction:

The video data can be protected against the channel errors by using the FEC schemes, which improve the successful data transmission probability and eliminate the costly retransmissions. An FEC code that provides *unequal error protection* (UEP) (i.e., the FEC code rates adaptive to the slice priority) can achieve considerable quality improvement compared to the *equal error protection* (EEP) FEC codes [23, 56, 57]. Recently, some schemes have also applied the FEC schemes both at APP PHY layers [54, 55, 58-63]. These schemes use the EEP or UEP FEC codes at APP and EEP codes at PHY. However, to the best of our knowledge, the cross-layer design of UEP FEC codes at both APP and PHY layers has not been investigated for prioritized video transmission.

For the cross-layer design of FEC codes at both layers, we address the three issues: (i) Since both FEC codes share a common channel bandwidth to add their redundancy, the optimal ratio of overhead added by each needs to be determined for a given channel SNR and bandwidth; (ii) We use the *systematic Raptor* codes [64 – 66] at APP and the RCPC codes [39] at PHY; (iii) To minimize the video distortion and maximize the video PSNR at a given channel bit rate and SNR, we perform a cross-layer optimization to find the optimal parameters of both FEC codes by considering the relative priorities of video packets. We assume that the channel SNR is obtained from the receiver in the form of channel side information (CSI) [10, 11, 54, 67, 68]. Our scheme provides higher transmission reliability to the high priority video slices at the expense of the

higher loss rates for low priority slices, and may also discard some low priority slices to meet the channel bit-rate limitations. We show that adapting the FEC code rates to the slice priority reduces the overall expected video distortion at the receiver. Our scheme does not assume retransmission of lost slices.

#### **4.1.1 Contributions**

Our proposed scheme is inspired by [55] and makes the following three contributions: *First*, the Raptor codes are generally used to provide EEP at APP. We use the systematic Raptor codes with a probability selection model to provide UEP for prioritized video data at APP. *Second*, we propose a cross-layer UEP FEC scheme using systematic Raptor codes at APP and RCPC codes [39] at PHY. To the best of our knowledge, no previous work exist on cross-layer UEP scheme at APP and PHY. We also compare the performance of the proposed UEP scheme with three other cross-layer FEC schemes. *Third*, we use a genetic algorithm (GA) based optimization of the proposed cross-layer FEC scheme, to maximize the video quality at the receiver for the AWGN and Rayleigh fading channels and a given bandwidth. The results demonstrate that our proposed cross-layer UEP scheme provides much better video quality than the other three FEC schemes.

#### **4.2 Related Work**

Several FEC coding schemes have been proposed at APP and PHY to provide UEP over AWGN channels [23, 30, 62, 69, 70] and fading channels [71-73]. Recently, the digital Fountain codes (also called rateless codes) have been used for forward error correction at APP. They can theoretically produce infinite number of encoded symbols from the source symbols. Luby [74]

developed the first practical class of rateless codes - Luby Transform (LT) codes. Shokrollahi [64] further extended the LT codes to Raptor codes. The Raptor codes have the following properties compared to the LT codes [64, 65]: (i) Raptor codes have linear encoding and decoding time, while the time complexity for LT codes is  $O(K \ln(K))$ , where  $K$  is the number of source symbols and  $\ln(K)$  is the average degree of symbols in a sparse graph; (ii) it is possible that some source symbols are not encoded and can therefore never be recovered in LT codes, whereas the design of Raptor codes ensures that each source symbol is encoded at least once. Due to their high recovery rate and low time complexity, the Raptor codes have been included in the Third Generation Partnership Project (3GPP) [66] and Digital Video Broadcasting (DVB) standard [75]. Detailed description of the Raptor codes can be found in [66, 75].

Kushwaha et al. [76] used LT codes to encode GOP of each layer of H.264 SVC video for transmission over cognitive radio wireless networks. Ahmad et al. [67] took advantage of the ratelessness of LT codes and proposed an adaptive FEC scheme for video transmission over Internet by employing feedback from receivers in the form of acknowledgement. Cataldi et al. [68] proposed sliding-window Raptor codes, which have a higher coding efficiency than the regular LT codes. They used these codes to provide UEP for a two-layer H.264/SVC scalable video. LT codes were also used in [77, 78] to design the streaming schemes with lower complexity. In [79], the authors proposed a combination of both packet-level and byte-level FEC to recover the errors in a multicast system. Zhang et al. [80] investigated how to optimally allocate rate among source, FEC and automatic repeat request (ARQ) for scalable video delivery over 3G wireless network.



In [55], the cross-layer design of FEC codes was studied at both layers for H.264 video transmission over AWGN channels. The UEP Luby transform codes were used at the APP and RCPC codes at the PHY. Stockhammer et al. [54] defined the protocol stack, including the FEC coding at APP and PHY, for the multimedia broadcast multicast service (MBMS) download and streaming in UMTS. A Raptor code was used at APP and the turbo code at PHY. Gomez and Bria [58] suggested employing the Raptor codes as APP FEC in DVB-H systems for mobile terminals and demonstrated its advantages over conventional multi-protocol encapsulation (MPE) FEC. Conventional MPE FEC employs the Reed-Solomon (RS) codes to encode the video stream; hence, it lacks the flexibility of LT coding at APP. Courtade and Wesel [59] considered a setup with LT coding at APP and FEC coding at PHY, and showed that the available channel bandwidth should be optimally split between APP and PHY FEC codes to improve the system performance.

Luby et al. [60] also considered employing two layers of EEP FEC at APP and PHY for MBMS download delivery in UMTS. They investigated the tradeoff between the APP FEC and PHY FEC codes, and studied the advantages of APP FEC on the system performance. Munaretto et al. [62] proposed an interesting optimization of APP FEC coding, video source coding, and PHY rate selection to improve the PSNR of delivered video on cellular networks. Authors in [63] also considered employing the Raptor codes at APP to improve the quality of service for video in MBMS in *long term evolution* (LTE) networks. They investigated the benefits of APP FEC to multicast multimedia contents and examined how much FEC redundancy should be used under different packet loss patterns.

## 4.3 Methods, Assumptions, and Procedures

### 4.3.1 Cross-Layer UEP using FEC Codes for Video Transmission

In this section, we discuss a priority assignment scheme for H.264/AVC video slices, design of UEP Raptor and RCPC codes, and our proposed cross-layer FEC scheme. We assume a unicast video transmission from a transmitting node to a destination node in a single hop wireless network, and ignore the intermediate network layers, i.e., transport, network, and link layers. This allows our algorithm to be generally applicable with different network protocol stacks.

#### 4.3.1.1 Priority Assignment for H.264 Video Slices

In H.264/AVC, the video frames are grouped into GOPs, and each GOP is encoded as a unit. We use a fixed slice size configuration where macroblocks of a frame are aggregated to form a fixed slice size. Let  $N_s$  be the average number of slices in one second of the video. More details of the video encoding parameters are given in Section 4.4.

H.264 slices can be prioritized based on their distortion contribution to the received video quality [22, 23, 56, 81]. In this scheme, all slices in a GOP are distributed into four priority classes of equal size based on their CMSE values, computed using Equation 1 in Section 3.3.1.1. The Priority 1 (Priority 4) slices introduce the highest (lowest) distortion to the received video quality. Note that using more than four slice priorities would generally result in a more accurate and flexible UEP coding at the cost of higher complexity due to a larger number of design

parameters. On the other hand, using less than four priority levels would limit the flexibility of our scheme and may decrease its performance [23, 55].

Let  $CMSE_i$  denote the average CMSE of all slices in a priority class  $i$ . We have

$CMSE_1 > CMSE_2 > CMSE_3 > CMSE_4$ . Since  $CMSE_i$  may vary considerably for various videos

depending on their spatial and temporal content, we use the normalized  $CMSE_i$ ,

$\overline{CMSE_i} = \frac{CMSE_i}{\sum_{i=1}^4 CMSE_i}$  to represent the relative importance of slices in a priority class [55]. In Table

3, we show  $CMSE_i$  for nine H.264 test video sequences, which have widely different spatial and temporal contents.

Table 3: Normalized  $CMSE, \overline{CMSE_i}$ , for Slices in Different Priorities of Sample Videos

Sequence	$\overline{CMSE_1}$	$\overline{CMSE_2}$	$\overline{CMSE_3}$	$\overline{CMSE_4}$
Coastguard	0.61	0.22	0.12	0.05
Foreman	0.63	0.21	0.11	0.05
Bus	0.64	0.21	0.10	0.04
Football	0.65	0.21	0.10	0.04
Silent	0.68	0.20	0.09	0.03
Woods	0.62	0.21	0.12	0.05
Whale Show	0.69	0.17	0.10	0.04
Stefan	0.61	0.24	0.12	0.03
Akiyo	0.85	0.12	0.03	0.01

In Table 3, first eight videos, which have very different characteristics (such as slow, moderate, and high motion), have almost similar  $\overline{\text{CMSE}}_i$  values. We also observed similar  $\overline{\text{CMSE}}_i$  values for other video sequences, such as Table Tennis and Mother Daughter. However, Akiyo, which is an almost static sequence with very little motion or scene changes, has different  $\overline{\text{CMSE}}_i$  values than other sequences. The  $\overline{\text{CMSE}}_i$  values changed only slightly when these videos were encoded at different bit rates (i.e., 512 Kbps and 1 Mbps) and slices sizes (150 bytes to 900 bytes). When these videos are encoded at 840 Kbps with 150 byte slices, we get  $N_s \approx 700$ . We choose the  $\overline{\text{CMSE}}_i$  values of Bus, which are similar to most other videos discussed above, to tune our proposed cross-layer scheme for all videos in Section 4.3.2. Since the  $\overline{\text{CMSE}}_i$  values of Akiyo are different, we also study the performance of the proposed cross-layer FEC scheme for Akiyo by using its own  $\overline{\text{CMSE}}_i$  values, and compare it with the performance of the scheme designed using the  $\overline{\text{CMSE}}_i$  values of Bus in Section 4.4.

#### 4.3.1.2 Design of UEP Raptor Codes at APP

The Raptor codes consist of a pre-code (e.g., a LDPC code) as the outer code and a weakened LT code as the inner code [64, 65]. They can be parameterized by  $(K, C, \Omega(x))$ , where

$K$  is the number of source symbols,  $C$  is a pre-code with block-length  $L$  and dimension  $K$ , and

$\Omega(x)$  is degree distribution of LT codes. Each encoded symbol is associated with an ID (ESI).

The pre-code and LT code can ensure a high decoding probability with a small coding overhead.

We use the systematic Raptor codes at APP [65, 66]. If there are  $K$  source symbols  $S[i]$  in one block,  $i = 0, \dots, K - 1$ , the first  $K$  encoded symbols are constructed such that  $E[0] = S[0], E[1] = S[1], \dots, E[K - 1] = S[K - 1]$ . The systematic Raptor codes can therefore correctly decode some source symbols even if the number of received encoded symbols  $N_r$  is less than the number of source symbols  $K$  [65].

The decoding failure probability of Raptor codes (i.e., the probability of at least one source symbol is not recovered) can be estimated as a function of  $K$  and  $N_r$  [54]:

$$P_e(\varepsilon_r) = \begin{cases} 1 & \text{if } \varepsilon_r < 0 \\ 0.85 \times 0.567^{\varepsilon_r} & \text{if } \varepsilon_r \geq 0 \end{cases} \quad (17)$$

where  $\varepsilon_r = N_r - K$  is the received encoding overhead of Raptor codes.

The average received overhead  $\bar{R}_{\varepsilon_r}$  to recover  $K$  source symbols can be calculated as [54]:

$$\bar{R}_{\varepsilon_r} \approx \frac{2}{K} \quad (18)$$

The number of additional encoded symbols needed for successfully decoding all the  $K$  source symbols is  $n = \bar{R}_{\epsilon_r} \times K \approx 2$ , which is independent of  $K$ . From (18), we also observe that the needed overhead (in percentage) for full symbol recovery decreases with the increase in  $K$ .

The Raptor codes are generally used to provide EEP at APP. We modify the Raptor codes with a probability selection model to provide UEP for video data at APP. Fig. 9 shows the framework of the proposed UEP Raptor encoder. To implement UEP with Raptor codes, we should generate more (less) coding overhead for higher (lower) priority symbols in order to provide higher (lower) level of protection to them. Assume we assign  $M$  priorities to video slices, where  $L_1$  is the highest priority, followed by  $L_2$ , and so on. If we have  $K_i$  source symbols (i.e., video slices) with priority  $L_i$ , we have  $K = \sum_{i=1}^M K_i$ . Let  $\rho_i = \frac{N_i}{N}$  for  $i = 1, \dots, M$  be the percentage of encoded symbols associated with data of priority level  $L_i$ .

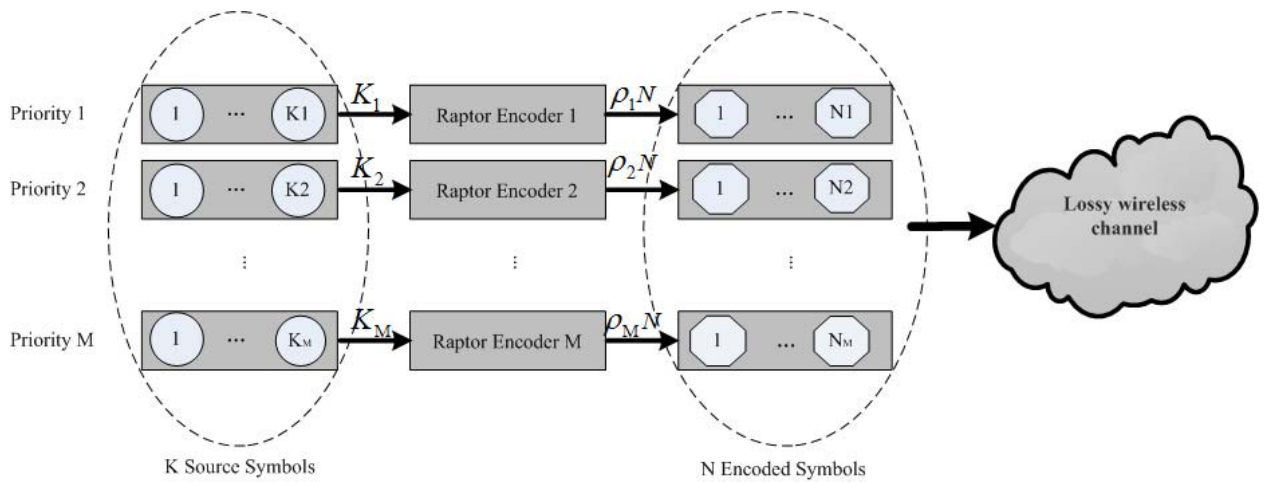


Figure 9: The framework of our proposed Raptor encoder.

We can get the lower bound of the symbol recovery rate  $P_{srr}$ , assuming a uniform channel symbol loss rate (PER):

$$\begin{aligned} P_{srr}(\varepsilon_r) &= P_e(\varepsilon_r) \times P_{r|F} + (1 - P_e(\varepsilon_r)) \times P_{r|S} \\ &= 0.85 \times 0.567^{\varepsilon_r} \times (1 - PER) + (1 - 0.85 \times 0.567^{\varepsilon_r}) \times 1 \end{aligned} \quad (19)$$

where  $P_{r|F}$  is the lower bound of symbol recovery rate when the complete decoding fails, and  $P_{r|S}$  is the symbol recovery rate when the complete decoding succeeds. In our system, we first assign the encoding overhead to the highest priority video slices, such that their recovery rate is above a predefined threshold  $\zeta_{r_1}$ . The remaining overhead is assigned to the lower priority video slices.

The minimum coding overhead  $\bar{R}_{\varepsilon_t}(K_i)$  for complete recovery of  $K_i$  source symbols of priority  $L_i$  with probability  $P_{srr}(\varepsilon_r(K_i))$  is given by

$$\bar{R}_{\varepsilon_t}(K_i) = \frac{K_i \times PER + \varepsilon_r(K_i)}{(1 - PER) \times K_i}, i = 1, \dots, M \quad (20)$$

where  $\varepsilon_r(K_i)$  is the required number of additional received symbols for priority class  $i$  in order to completely recover the source symbols of this priority.

#### 4.3.1.3 Design of RCPC Codes at PHY

We use RCPC codes at PHY because of their flexibility in providing various code rates. RCPC codes use a low-rate convolutional mother code with various puncturing patterns to obtain

codes with various rates. The RCPC decoder employs a Viterbi decoder, whose bit error rate (BER)  $P_b$  is upper bounded by [39]

$$P_b \leq \frac{1}{P} \sum_{d=d_f}^{\infty} c_d P_d \quad (21)$$

where  $d_f$  is the free distance of the convolutional code,  $P$  is the puncturing period, and  $c_d$  is the total number of error bits produced by the incorrect paths and is known as the distance spectrum. Finally,  $P_d$  is the probability of selecting a wrong path in Viterbi decoding with Hamming distance  $d$ .  $P_d$  depends on the modulation and channel characteristics.

For an RCPC code with rate  $R$ , using the AWGN channel, BPSK modulation and the symbol to noise power ratio  $\frac{E_S}{N_0} = R \frac{E_b}{N_0}$ , the value of  $P_d$  (using soft Viterbi decoding) is given by

$$P_d = \frac{1}{2} \operatorname{erfc} \sqrt{d \frac{E_S}{N_0}} = Q \sqrt{2d \frac{E_S}{N_0}} \quad (22)$$

where  $Q(\lambda) = \frac{1}{\sqrt{2\pi}} \int_{\lambda}^{\infty} e^{-\frac{t^2}{2}} dt$ .



For an RCPC code with rate  $R$ , using a Rayleigh flat fading channel with perfect channel estimation and soft decision decoding, BPSK modulation and the symbol to noise power ratio

$\frac{E_S}{N_0} = R \frac{E_b}{N_0}$ , the value of  $P_d$  (using soft Viterbi decoding) is given by [82]

$$P_d = q^d \sum_{k=0}^{d-1} \binom{d-1+k}{k} (1-q)^k \quad (23)$$

where  $q = \frac{1}{2} (1 - \sqrt{\frac{\bar{\gamma}R}{(1+\bar{\gamma}R)}})$  and  $\bar{\gamma} = \frac{E_b}{N_0}$ .

At PHY, the cyclic redundancy check (CRC) bits are added to each APP-frame to detect RCPC decoding error(s). We use the CRC-8 given by the polynomial  $1 + x^2 + x^4 + x^6 + x^7 + x^8$  [83]. Next, each APP-frame is encoded using an RCPC code, with

the mother code rate of  $R = \frac{1}{3}$  and memory  $M = 6$ . For four priority groups of APP-frames, we

have  $R_1 \leq R_2 \leq R_3 \leq R_4$  and  $R_i \in \{\frac{8}{8}, \frac{8}{9}, \frac{8}{10}, \frac{8}{12}, \frac{8}{14}, \frac{8}{16}, \frac{8}{18}, \frac{8}{20}, \frac{8}{22}, \frac{8}{24}\}$  where  $R_i$  represents the

RCPC code rate of priority  $i$  APP-frames. Therefore, the parameters that need to be tuned at

PHY are  $R_1$  through  $R_4$ . We refer to a APP-frame encoded by the RCPC code as a PHY-frame.

Without the loss of generality, we assume that each transmitted packet contains one PHY-frame. Note that the number of PHY-frames in a packet does not affect the optimum cross-layer setup of FEC codes in our scheme. We have used a conventional BPSK modulation, and AWGN

and Rayleigh flat fading channels. However, our model can be easily extended to more complex channel models by using an appropriate  $P_d$  in (23). Recently, several FEC coding schemes have been proposed at APP and PHY to provide UEP over AWGN channels [23, 30, 62, 69, 70] and fading channels [71-73].

#### 4.3.1.4 System Model at Transmitter

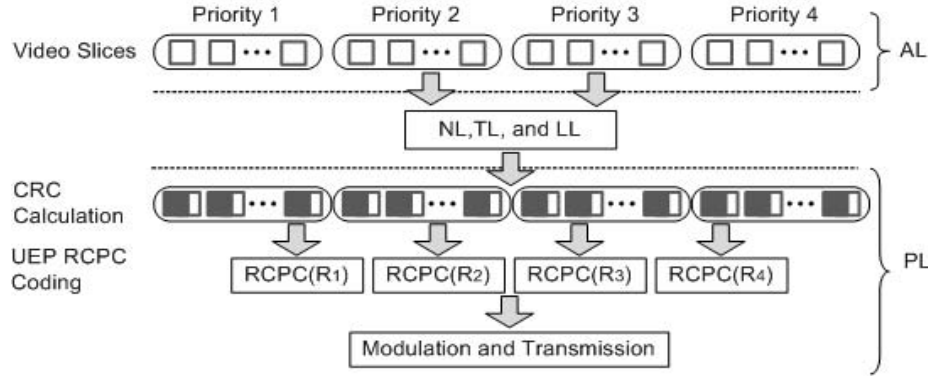
Based on our discussions so far, we use four combinations of cross-layer FEC coding schemes at APP and PHY as summarized in Table 4. For protecting the data against wireless channel errors, the FEC coding is necessary at PHY but optional at APP. Fig. 10(a) and 10(b) illustrate these cross-layer FEC schemes. The cross-layer optimization of these FEC-schemes is discussed in Section 4.4.1.

Table 4: Various Combinations of Cross-Layer FEC Coding Schemes

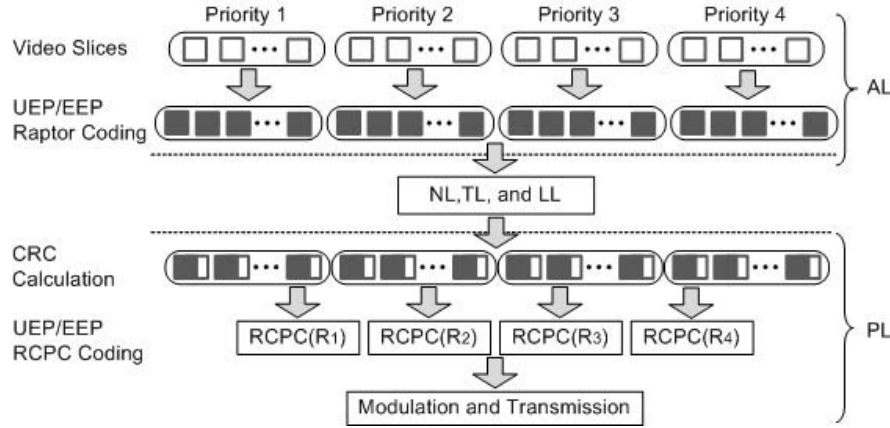
Model	S-I	S-II	S-III	S-IV
APP FEC	No FEC	EEP	UEP	UEP
PHY FEC	UEP	UEP	EEP	UEP

In S-I scheme, the FEC coding is applied only at PHY to protect the video slices based on their priority by using the UEP RCPC coding. The priority of each APP-frame is conveyed to PHY by using cross-layer communication. This scheme is similar to the FEC schemes proposed in [23, 30, 56, 57, 70, 84, 85]. The S-II, S-III, and S-IV schemes represent the cross-layer FEC schemes where video data is protected at both APP and PHY. In S-II scheme, the regular systematic Raptor codes and UEP RCPC codes are applied at APP and PHY, respectively. The S-III scheme applies UEP Raptor and EEP RCPC code at APP and PHY, respectively. The S-II and S-III schemes are similar to the FEC schemes proposed in [54, 55, 58-63, 86], in which EEP

or UEP FEC codes are used at APP and EEP codes at PHY. In S-IV scheme, the UEP Raptor codes and UEP RCPC codes are applied at APP and PHY, respectively. To the best of our knowledge, no such cross-layer FEC scheme (i.e., S-IV) is available in the literature.



(a) S-I FEC scheme; video slices are prioritized at APP and UEP FEC coding is applied at PHY. Here, TL, NL, and LL represent the transport, network, and link layers, respectively.



(b) S-II, S-III and S-IV cross-layer FEC schemes. In these schemes, a cross-layer FEC coding is applied with EEP (or UEP) Raptor coding at APP and EEP (or UEP) RCPC coding at PHY. For EEP at PHY, code rates are  $R_1 = R_2 = R_3 = R_4 = R$ .

Figure 10: Illustration of four cross-layer FEC schemes.

#### 4.3.1.5 Decoding at Receiver

Let  $PER_i$  denote the packet error rate of APP-frames of priority  $i$  at the receiver after RCPC decoding and before Raptor decoding at APP.  $PER_i$  can be computed by using BER from (21).

In S-I scheme, each APP-frame consists of an uncoded video slice as the Raptor coding is not applied at APP. Therefore, the *video slice loss rate* (VSLR) of source packets with priority  $i$  is  $VSLR_i = PER_i$ . In S-II through S-IV schemes, the Raptor coding is also applied and the decoding error rate of Raptor codes should be considered in  $VSLR_i$ . In S-III scheme, the EEP RCPC code is used at PHY, hence we have  $PER_1 = PER_2 = PER_3 = PER_4 = PER$ . In S-II and S-IV schemes,  $PER_1 \leq PER_2 \leq PER_3 \leq PER_4$  since the UEP RCPC codes are applied at PHY. If the Raptor codes are used at APP, we employ (19) to find the final Raptor decoding symbol recovery rates  $P_{srr}(i), i \in 1, \dots, 4$  for each priority at the receiver (see Section 4.3.1.2). If the symbol recovery rate of priority  $i$  is  $P_{srr}(i)$ , then  $VSLR_i = 1 - P_{srr}(i)$ .

#### 4.3.2 Cross-Layer Optimization of FEC Codes

In our cross-layer FEC schemes, the APP and PHY FEC codes share the same available channel bandwidth. As the channel SNR increases, the RCPC code rate at PHY can be increased,

and more channel bandwidth becomes available for Raptor coding at APP. For low channel SNR, assigning a higher portion of the available redundancy to Raptor codes at APP may not improve the delivered video quality since almost all PHY-frames would be corrupted during transmission. Therefore, a lower RCPC code rate should be used at PHY, which would consume a larger portion of the channel bandwidth allowing only a weaker Raptor code at APP.

We discuss below the optimization to find the optimal parameters for the FEC schemes.

#### 4.3.2.1 Formulation of Optimization Problem

The goal of cross-layer optimization in our scheme is to deliver a video with the highest possible PSNR for a given channel bandwidth  $C$  and SNR. Since computing the video PSNR requires decoding the video at the receiver, it is not feasible to use PSNR directly as the optimization metric due to its heavy computational complexity. Therefore, we use a low-complexity substitute function  $F$  to represent the behavior of video PSNR.

The PSNR of a video stream depends on the percentage of lost slices and their CMSE values [22, 23]. However, the slice loss may not be linearly correlated to the decrease in PSNR. Therefore, we use a function "normalized F", denoted by  $\bar{F}$ , to capture the behavior of PSNR based on the slice loss rates and their CMSE as follows [55]:

$$\bar{F} = \sum_{i=1}^r \overline{CMSE}_i^\alpha \cdot VSLR_i \quad (24)$$

Here  $r$  is the number of slice priorities and  $(\overline{CMSE_i})$  is the normalized CMSE value which represents the relative priority (i.e., weight) of priority  $i$  slices. The parameter  $\alpha \geq 0$  adjusts the *weight* assigned to slices of each priority level such that minimizing  $F$  results in maximizing the video PSNR; In [55], the optimal value of  $\alpha$  was found to be 1.

To minimize  $F$ , we tune the parameters of the FEC codes at APP and PHY. In S-I, the optimization parameters are  $R_1$  through  $R_4$ , such that  $R_1 \leq R_2 \leq R_3 \leq R_4$ . For this scheme, the optimization function can be written as

$$\begin{aligned} \{R_1^*, R_2^*, R_3^*, R_4^*\} &= \underset{\{R_1, R_2, R_3, R_4\}}{\operatorname{argmin}} \bar{F}, \\ \text{s.t. } \sum_{i=0}^4 N_s(S+1) \frac{R_i^{-1}}{4} &\leq C. \end{aligned} \quad (25)$$

where  $S+1$  is the slice size  $S = 150$  bytes plus one byte CRC.

In S-II, the UEP RCPC codes at PHY and EEP Raptor codes at APP are used, and the optimization parameters are  $R_1$  through  $R_4$ , and  $\theta_t$ . Here  $\theta_t$  is the Raptor coding overhead, which is slightly greater than one. Hence, the Raptor encoder will generate  $\theta_t N_s$  encoded symbols. The number of encoded symbols generated by Raptor encoder for each priority is

$N_i = \rho_i \theta_t N_s, i \in 1, \dots, 4$  and  $\sum_{i=0}^4 N_i = \theta_t N_s$ . Since EEP FEC is used at APP, we have

$\rho_1 = \rho_2 = \rho_3 = \rho_4 = 0.25$ . As a result, the optimization function is

$$\begin{aligned} \{R_1^*, R_2^*, R_3^*, R_4^*, \theta_t^*\} &= \underset{\{R_1, R_2, R_3, R_4, \theta_t\}}{\operatorname{argmin}} \bar{F}, \\ \text{s.t. } \sum_{i=0}^4 \theta_t N_s (S+1) \frac{R_i^{-1}}{4} &\leq C. \end{aligned} \quad (26)$$

In S-III, UEP Raptor codes at APP and EEP RCPC codes at PHY are used, and the optimization parameters are  $\rho_1$  through  $\rho_3$ ,  $\theta_t$ , and  $R$ . Here, the value of  $\rho_4$  can be determined

based on  $\rho_1$  through  $\rho_3$  since  $\sum_{i=0}^4 \rho_i = 1$ . As a result, the optimization function is

$$\begin{aligned} \{\rho_1^*, \rho_2^*, \rho_3^*, \theta_t^*, R^*\} &= \underset{\{\rho_1, \rho_2, \rho_3, \theta_t, R\}}{\operatorname{argmin}} \bar{F}, \\ \text{s.t. } \theta_t N_s (S+1) R^{-1} &\leq C. \end{aligned} \quad (27)$$

In S-IV, UEP FEC codes are used at both layers, and optimization parameters are  $\rho_1$  through  $\rho_3$ ,  $\theta_t$ , and  $R_1$  through  $R_4$ . The optimization function is

$$\begin{aligned} \{\rho_1^*, \rho_2^*, \rho_3^*, \theta_t^*, R_1^*, R_2^*, R_3^*, R_4^*\} &= \underset{\{\rho_1, \rho_2, \rho_3, \theta_t, R_1, R_2, R_3, R_4\}}{\operatorname{argmin}} \bar{F}, \\ \text{s.t. } \sum_{i=0}^4 \rho_i \theta_t N_s (S+1) R_i^{-1} &\leq C. \end{aligned} \quad (28)$$

The optimization of Raptor code parameters involves employing (19) for various priority levels, which cannot be represented by a linear function. Also, the concatenation of two FEC codes presents a nonlinear optimization problem. We use the genetic algorithms (GA) toolbox available in Matlab [87] to perform optimizations, as GA can give solutions which are close to the global optimum [88-90]. For performance evaluation of GA methods, we refer the interested readers to [89, 91].

In Table 3, the normalized CMSE values ( $\overline{CMSE_i}$ ) of the video sequences, except Akiyo, were similar. Therefore, the optimal parameters computed for Bus video would be almost optimal for the other four video sequences generated by the same encoding parameters. We therefore use the  $\overline{CMSE_i}$  of the Bus video with data rate of 840 Kbps to perform our optimizations. We implement our cross-layer FEC setup for S-I through S-IV (see Table 4) in Matlab environment.

#### 4.4 Results and Discussion

In this section, we evaluate the performance of our optimized cross-layer FEC schemes for four CIF ( $352 \times 288$  pixels) test video sequences, Bus, Foreman, Coastguard, and Akiyo. These sequences have different texture and motion contents. A frame of these test video sequences is shown in Fig. 11. These sequences were encoded using H.264/AVC JM 14.2 reference software [92] at 840 Kbps and 150 bytes slice size, for a GOP length of 30 frames with GOP structure



*IDR B P B ...P B* at 30 frames/sec. The slices were formed using dispersed mode FMO with two slice groups per frame. Two reference frames were used for predicting the *P* and *B* frames, with error concealment enabled using temporal concealment and spatial interpolation.



Foreman



Bus



Akiyo

Figure 11: A frame of three test video sequences.

Table 5: Optimum Cross-Layer Parameters for S-I Scheme, at C = 1.4 Mbps

$E_s/N_0$	-1 dB	-0.5 dB	0 dB	0.5 dB	1 dB	1.5 dB	2 dB	2.5 dB	3 dB
$\bar{F}$	0.366	0.244	0.178	0.125	0.065	0.034	0.009	0.002	0
$F_{\text{Bus}}$	162.3	108.3	77.4	55.4	29.5	14.6	3.9	0.7	0.1
$F_{\text{Coastguard}}$	67.5	44.7	31.5	23.6	12.7	6.1	1.6	0.3	0
$F_{\text{Foreman}}$	70.0	46.0	31.6	23.9	11.7	6.6	1.7	0.3	0
$R_1$	8/24	8/20	8/18	8/16	8/16	8/14	8/14	8/14	8/14
$R_2$	8/12	8/16	8/18	8/14	8/14	8/14	8/14	8/14	8/14
$R_3$	8/8	8/8	8/8	8/14	8/14	8/12	8/12	8/12	8/12
$R_4$	8/8	8/8	8/8	8/8	8/8	8/12	8/12	8/12	8/12
$VSLR_1$	0.009	0.032	0.027	0.030	0.007	0.016	0.004	0	0
$VSLR_2$	1	0.350	0.027	0.206	0.063	0.016	0.004	0	0
$VSLR_3$	1	1	1	0.206	0.063	0.137	0.036	0.009	0.002
$VSLR_4$	1	1	1	1	1	0.137	0.036	0.009	0.002

We have used two channel transmission rates of C = 1.4 Mbps and C = 1.8 Mbps to study the performance over AWGN channels and a channel transmission rate of C = 1.4 Mbps over Rayleigh flat fading channels. The video slices are prioritized into four priority levels as discussed in Section 4.3.1.1. Video slices of each priority level are encoded by independent Raptor encoders so that their priorities are maintained and can be used by the RCPC code at PHY. For different channel SNRs, appropriate selection probabilities for Raptor codes are chosen to provide UEP based on the normalized slice CMSE values.

#### 4.4.1 Discussion of Cross-Layer Optimization Results

We present the cross-layer optimization results, including the FEC parameters (e.g.,  $R_i$  for RCPC, and  $\theta_i$  and  $\rho_i$  for Raptor codes),  $VSLR_i$ ,  $\bar{F}$ , and  $F$ . Here  $F$  is calculated by replacing the  $\overline{CMSE}_i$  by the actual average  $CMSE_i$  in (24), for the H.264 encoded video sequence under consideration. We first evaluate the performance of the cross-layer FEC schemes over AWGN channels. The experiments for the fading channel are discussed in Section 4.4.3. We use one GOP of video data as a source block to be encoded by Raptor codes and the optimum FEC code rates are computed for slices of each GOP according to the average channel SNR. The results of all four FEC schemes for three test video sequences (Bus, Foreman and Coastguard), encoded at 840 Kbps, are reported in Tables 5 through 8 for channel bit rate  $C = 1.4$  Mbps. Fig. 12(a) and 12(b) show the minimum normalized  $\bar{F}$  achieved by the optimized cross-layer schemes for the two channel bit rates. The results for Akiyo video sequence are discussed in Section 4.4.2. For a GOP length of 30 frames (corresponding to 1 second video duration at 30 frames/second), the optimization process takes about 50 ms in Matlab, on a Intel Core 2 Duo, 2.2 GHz, 3 GB RAM computer. For one or two video frames (instead of a whole GOP), the optimization process takes about 7 ms and 18 ms, respectively.

Table 6: Optimum Cross-Layer Parameter for S-II Scheme, C = 1.4 Mbps

Es/No	-1 dB	-0.5 dB	0 dB	0.5 dB	1 dB	1.5 dB	2 dB	2.5 dB	3 dB
$\bar{F}$	0.360	0.244	0.173	0.125	0.064	0.034	0	0	0
$F_{Bus}$	160.1	108.3	77.4	55.4	29.2	14.6	0.1	0	0
$F_{Coastguard}$	66.5	44.7	31.5	23.6	12.5	6.1	0	0	0
$F_{Foreman}$	68.6	46.0	31.6	23.9	11.6	6.6	0	0	0
$R_1$	8/24	8/20	8/18	8/16	8/16	8/14	8/12	8/12	8/10
$R_2$	8/8	8/16	8/18	8/14	8/14	8/14	8/12	8/12	8/10
$R_3$	8/8	8/8	8/8	8/14	8/14	8/12	8/12	8/12	8/10
$R_4$	8/8	8/8	8/8	8/8	8/8	8/12	8/12	8/12	8/10
$\theta_t$	1.14	1.01	1.01	1.01	1.01	1.01	1.10	1.10	1.31
$VSLR_1$	0	0.032	0.027	0.030	0.006	0.016	0	0	0
$VSLR_2$	1	0.350	0.027	0.206	0.063	0.016	0	0	0
$VSLR_3$	1	1	1	0.206	0.063	0.137	0	0	0
$VSLR_4$	1	1	1	1	1	0.137	0	0	0

Table 7: Optimum Cross-Layer Parameters for S-III Schemes, C = 1.4 Mbps

Es/No	-1 dB	-0.5 dB	0 dB	0.5 dB	1 dB	1.5 dB	2 dB	2.5 dB	3 dB
$\bar{F}$	1	1	0.971	0.766	0.092	0.015	0	0	0
$F_{Bus}$	407.9	407.7	396.1	312.4	42.0	7.0	0	0	0
$F_{Coastguard}$	180.1	180.0	174.9	137.9	17.3	3.0	0	0	0
$F_{Foreman}$	214.9	214.8	208.7	164.6	16.2	2.6	0	0	0
$R$	8/12	8/12	8/12	8/12	8/12	8/12	8/12	8/12	8/10
$\theta_t$	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.31

$\rho_1$	0.249	0.249	0.249	0.249	0.398	0.278	0.253	0.253	0.253
$\rho_2$	0.249	0.249	0.249	0.249	0.394	0.273	0.250	0.250	0.252
$\rho_3$	0.228	0.228	0.228	0.228	0.209	0.273	0.249	0.249	0.250
$\rho_4$	0.274	0.274	0.274	0.274	0	0.175	0.248	0.248	0.245
$VSLR_1$	1	1	0.971	0.766	0.002	0.001	0	0	0
$VSLR_2$	1	1	0.971	0.766	0.006	0.004	0	0	0
$VSLR_3$	1	1	0.971	0.766	0.447	0.004	0	0	0
$VSLR_4$	1	1	0.971	0.766	1	0.335	0	0	0

Table 8: Optimum Cross-Layer Parameters for S-IV Schemes, C = 1.4 Mbps

Es/No	-1 dB	-0.5 dB	0 dB	0.5 dB	1 dB	1.5 dB	2 dB	2.5 dB	3 dB
$\bar{F}$	0.150	0.150	0.116	0.040	0.039	0.015	0	0	0
$F_{Bus}$	68.3	68.3	53.1	18.7	18.2	6.8	0	0	0
$F_{Coastguard}$	27.5	27.4	21.6	8.1	7.9	3.0	0	0	0
$F_{Foreman}$	26.6	26.6	20.5	6.8	6.6	2.5	0	0	0
$R_1$	8/18	8/16	8/16	8/16	8/14	8/14	8/12	8/12	8/10
$R_2$	8/18	8/16	8/14	8/14	8/14	8/14	8/12	8/12	8/10
$R_3$	8/8	8/12	8/12	8/14	8/12	8/12	8/12	8/12	8/10
$R_4$	8/8	8/8	8/8	8/8	8/12	8/12	8/12	8/12	8/10
$\theta_t$	1.01	1.01	1.01	1.01	1.01	1.01	1.10	1.10	1.31
$\rho_1$	0.499	0.426	0.296	0.287	0.275	0.258	0.254	0.254	0.254

$\rho_2$	0.500	0.422	0.539	0.357	0.273	0.255	0.252	0.252	0.252
$\rho_3$	0.001	0.152	0.164	0.355	0.430	0.294	0.248	0.248	0.249
$\rho_4$	0	0	0.001	0.001	0.022	0.193	0.246	0.246	0.245
$VSLR_1$	0	0	0.001	0	0.001	0.001	0	0	0
$VSLR_2$	0	0	0.005	0	0.002	0.002	0	0	0
$VSLR_3$	1	1	0.682	0	0.002	0.007	0	0	0
$VSLR_4$	1	1	1	1	0.947	0.325	0.001	0	0

Since results for the three video sequences show the same trends, we discuss here the results only for Bus video. For  $E_s/N_0 \leq 1$  dB in Tables 5 to 8 and Fig. 12(a), the rank of different schemes based on the minimum  $\bar{F}$  is  $S\text{-IV} \geq S\text{-II} \geq S\text{-I} \geq S\text{-III}$  for channel bit rate  $C = 1.4$  Mbps. At low channel SNR, the use of UEP RCPC coding at PHY (in S-I) achieves much better performance than the use of EEP RCPC coding at PHY and UEP Raptor coding at APP (in S-III) because: (i) Many packets are corrupted in S-III as the EEP FEC codes at PHY cannot protect all of them effectively due to constrained channel bandwidth. (ii) The UEP RCPC code in S-I provides better protection to the higher priority slices. As a result, more higher priority slices are transmitted error-free than in S-III. (iii) The use of Raptor codes at APP (in S-III) is not helpful when many slices are corrupted at PHY as enough error-free source symbols are not received at APP. A similar behavior is observed in Fig. 12(b) for a relatively lower  $E_s/N_0 < -0.5$  dB at channel bit rate  $C = 1.4$  Mbps because more channel bandwidth is available to provide a stronger FEC protection in this case.

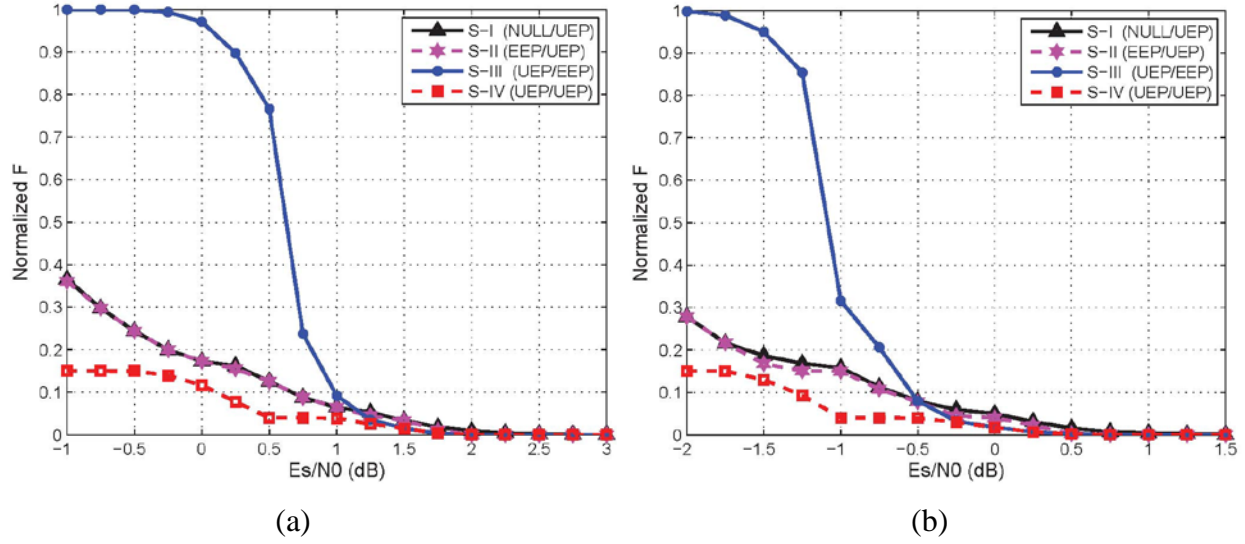


Figure 12: Normalized  $\bar{F}$  of Bus sequence for AWGN channel SNRs at channel bit rates:

(a)  $C = 1.4$  Mbps and (b)  $C = 1.8$  Mbps.

Another interesting observation for  $E_s/N_0 \leq 1$  dB at  $C = 1.4$  Mbps is that S-II (which uses UEP RCPC code at PHY and EEP Raptor code at APP) does not perform better than S-I scheme. This is because, for successful decoding of all the Raptor coded symbols, the number of received encoded symbols should be larger than the number of source symbols. For lower channel SNRs, assigning a higher portion of the available channel bandwidth to Raptor codes will not improve the delivered video quality since almost all PHY-frames would be corrupted during transmission. Therefore, the optimization algorithm assigns most of the available coding overhead to RCPC at PHY, while allowing a weaker Raptor code at APP, which decreases  $PER$ . As a result, the channel bandwidth available for the EEP Raptor codes at APP is not enough to successfully decode all the source symbols. For  $C = 1.8$  Mbps, Fig. 12(b) exhibits the same behavior for  $E_s/N_0 \leq 0$  dB.

The S-IV scheme, which uses UEP at both layers, achieves better performance than the other three schemes under all channel conditions. In this scheme, different slices are protected according to their priority at both layers. This scheme therefore benefits both from the ratelessness as well as the UEP property. For  $E_s/N_0 < 1$  dB at channel bit rate  $C = 1.4$  Mbps, the S-IV schemes achieves much better performance than other schemes because using UEP FEC codes at both layers provide stronger protection to higher priority video slices compared to the lower priority slices. Fig. 12(b) shows similar results for  $E_s/N_0 < -0.5$  dB at channel bit rate  $C = 1.8$  Mbps.

For  $E_s/N_0 \geq 1.5$  dB in Tables 5 to 8 and Fig. 12(a), the ranking of different schemes for achieving the minimum  $\bar{F}$  is  $S\text{-IV} \geq S\text{-III} \geq S\text{-II} \geq S\text{-I}$ . At higher channel SNR, fewer packets are corrupted at PHY and thus our optimization algorithm allocates more channel bandwidth to Raptor codes at APP. As a result, the UEP Raptor codes (in S-III and S-IV) achieve better performance than EEP Raptor codes (in S-II), followed by no FEC at APP (in S-I). Similar behavior is also observed for  $C = 1.8$  Mbps in Fig. 12(b) for  $E_s/N_0 \geq 0.5$  dB. As channel SNR increases further (i.e.,  $E_s/N_0 > 2.5$  dB) for channel bit rate  $C = 1.4$  Mbps, the difference of optimum  $\bar{F}$  between different schemes is negligible because very few packets are corrupted due



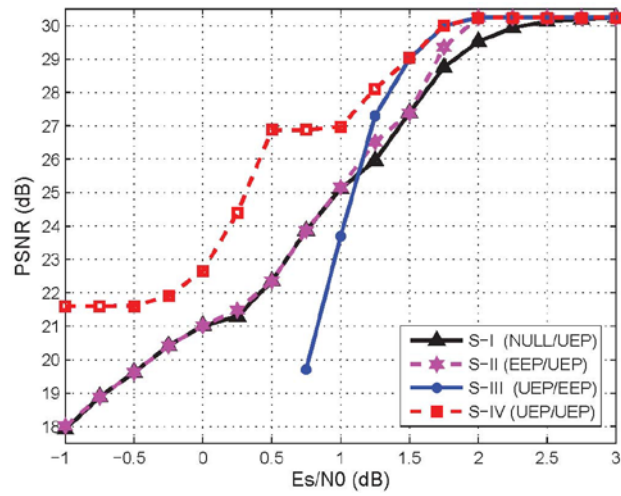
to channel error and the EEP FEC codes can provide enough protection. The same performance is achieved for  $E_s/N_0 > 1$  dB at channel bit rate  $C = 1.8$  Mbps.

Fig. 12(a) and 12(b) also reveal that FEC at APP is more effective for a channel with  $C = 1.8$  Mbps than for  $C = 1.4$  Mbps, especially when the channel SNR is low. For example, the S-III outperforms S-I and S-II schemes for  $\frac{E_s}{N_0} > -0.5$  dB at  $C = 1.8$  Mbps, whereas the same result is achieved for  $\frac{E_s}{N_0} > 1.0$  dB at  $C = 1.4$  Mbps. This is because more channel bandwidth is available in the former case that can be assigned to Raptor codes at APP to provide more protection to video data.

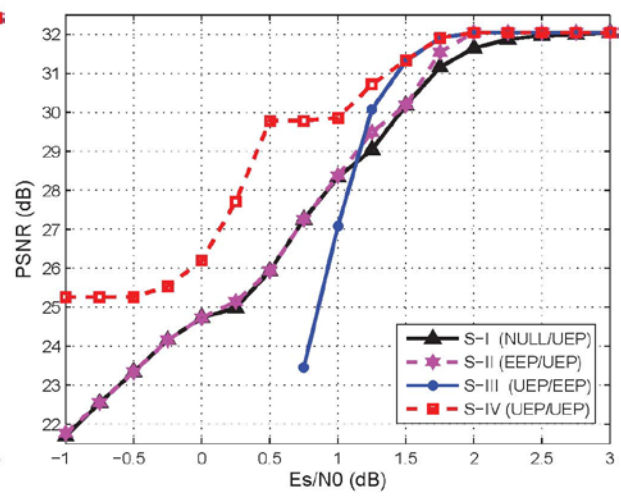
Overall, the proposed S-IV scheme achieves the best performance for all three video sequences under different channel SNRs and  $C$ . Therefore, we can generally conclude that cross-layer UEP provides best protection for video transmission among the four cross-layer schemes used in this section.

Note that the optimization is performed only once for a given set of  $\overline{CMSE_i}$  values, a GOP structure, and a set of channel SNRs, and need not to be run separately for each GOP. The same set of optimum parameters can be used for any video stream with the same GOP structure and similar CMSEs.

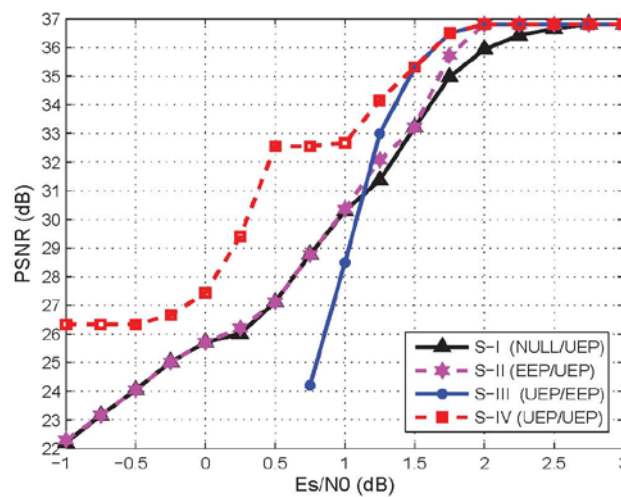
#### 4.4.2 Performance of Cross-Layer FEC Schemes for Test Videos over AWGN Channels



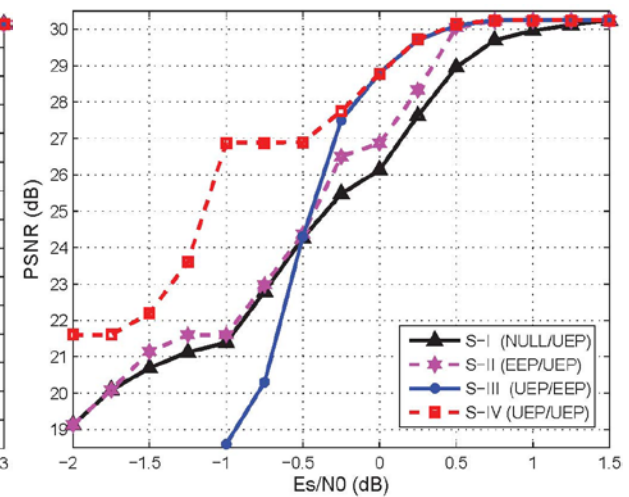
(a)



(b)



(c)



(d)

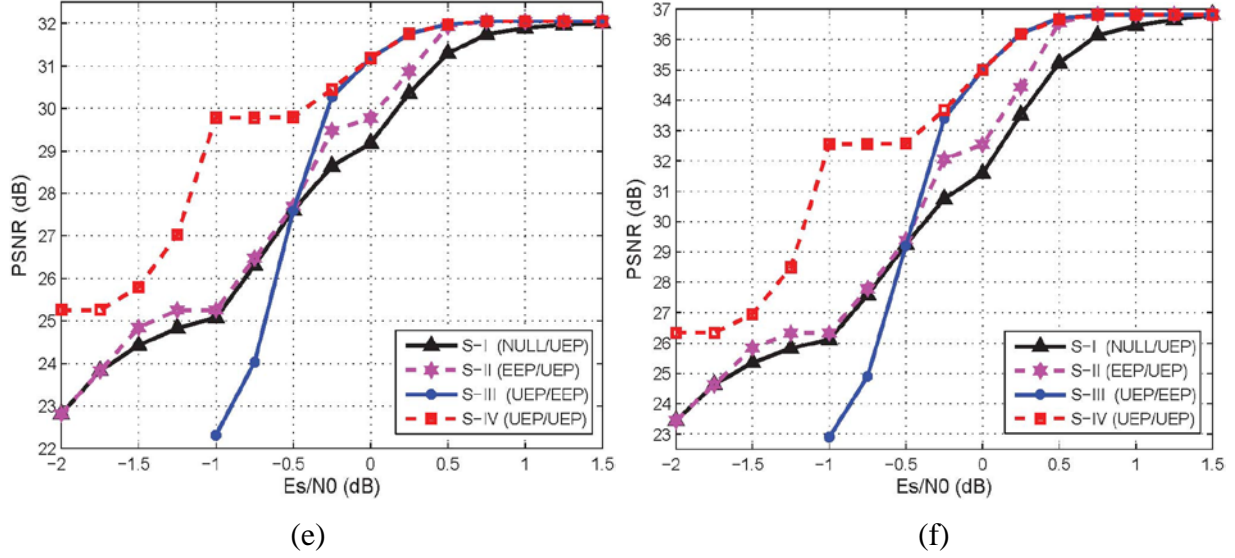


Figure 13: Average PSNR of test videos for different channel SNRs for AWGN channel: (a) Bus sequence at C=1.4 Mbps, (b) Coastguard sequence at C=1.4 Mbps, (c) Foreman sequence at C=1.4 Mbps, (d) Bus sequence at C=1.8 Mbps, (e) Coastguard sequence at C=1.8 Mbps, (f) Foreman sequence at C=1.8 Mbps. The PSNR of Bus, Coastguard, and Foreman at error-free channel are 30.24 dB, 32.05 dB, and 36.81 dB, respectively.

Table 9: Optimal Cross-Layer Parameters for S-IV at C = 1.4 Mbps for Akiyp Sequence

Es/No	-1 dB	-0.5 dB	0 dB	0.5 dB	1 dB	1.5 dB	2 dB	2.5 dB	3 dB
$F_{opt}$	1.052	1.051	0.802	0.194	0.192	0.077	0.001	0	0
$F_{sub}$	1.052	1.051	0.808	0.194	0.205	0.096	0.001	0	0
$PSNR_{opt}$	29.83	29.83	33.53	41.53	41.69	44.49	46.34	46.35	46.35
$PSNR_{sub}$	29.83	29.83	33.47	41.53	41.54	44.17	46.33	46.35	46.35
$R_1$	8/18	8/16	8/16	8/16	8/14	8/14	8/12	8/12	8/10
$R_2$	8/18	8/16	8/14	8/14	8/14	8/14	8/12	8/12	8/10
$R_3$	8/8	8/12	8/14	8/14	8/12	8/12	8/12	8/12	8/10
$R_4$	8/8	8/8	8/8	8/8	8/12	8/12	8/12	8/12	8/10
$\theta_t$	1.01	1.01	1.01	1.01	1.01	1.01	1.10	1.10	1.31

$\rho_1$	0.499	0.435	0.299	0.293	0.281	0.264	0.257	0.257	0.254
$\rho_2$	0.500	0.428	0.544	0.358	0.274	0.259	0.250	0.250	0.253
$\rho_3$	0.001	0.137	0.157	0.349	0.430	0.294	0.248	0.248	0.249
$\rho_4$	0	0	0	0	0.015	0.183	0.245	0.245	0.244
$VSLR_I$	0	0	0	0	0	0	0	0	0
$VSLR_I$	0	0	0.003	0	0.001	0	0	0	0
$VSLR_I$	1	1	0.693	0	0.002	0.007	0	0	0
$VSLR_I$	1	1	1	1	0.965	0.360	0.001	0	0

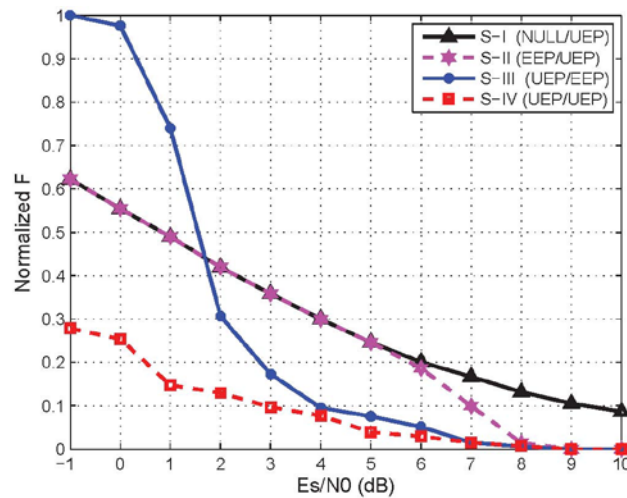
We used the slice loss rates reported in Tables 5 through 8 to evaluate the average PSNR of three video sequences (Bus, Foreman, and Coastguard) in Figures 13(a) through 13(c) for  $C = 1.4$  Mbps. Similarly, the slice loss rates were used to evaluate the average PSNR of these video sequences in Figures 13(d) through 13(f) for  $C = 1.8$  Mbps. From these figures, we observe that the PSNRs of the test videos are excellent match with the corresponding  $\bar{F}$  and  $F$  obtained by numerical optimization in Section 4.4.1.

Fig. 13 confirms that our proposed cross-layer FEC S-IV scheme, with UEP coding at APP and PHY, achieves considerable improvement in average video PSNR over the remaining three schemes. It outperforms S-I and S-II schemes by about  $1.5 \sim 4$  dB for  $\frac{E_s}{N_0} \leq 1.5$  dB, and S-III scheme by more than 3 dB for  $\frac{E_s}{N_0} \leq 1$  dB (at  $C = 1.4$  Mbps). At  $C = 1.8$  Mbps, S-IV

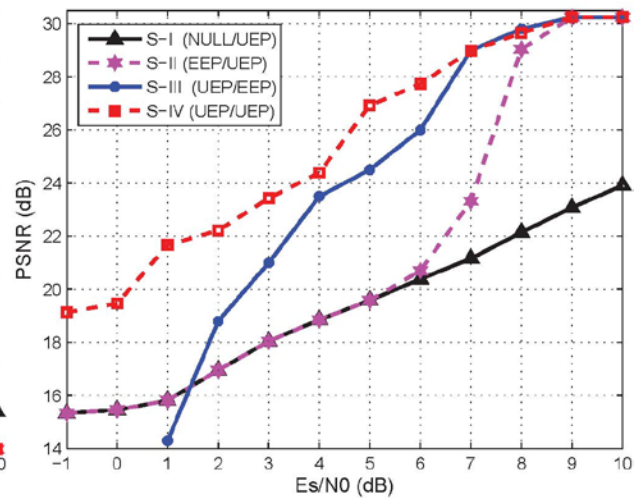
outperforms the S-I and S-II schemes by about 1~4 dB for  $\frac{E_s}{N_0} \leq 0.25$  dB, and the S-III scheme

by about 2~7 dB for  $\frac{E_s}{N_0} \leq -0.5$  dB.

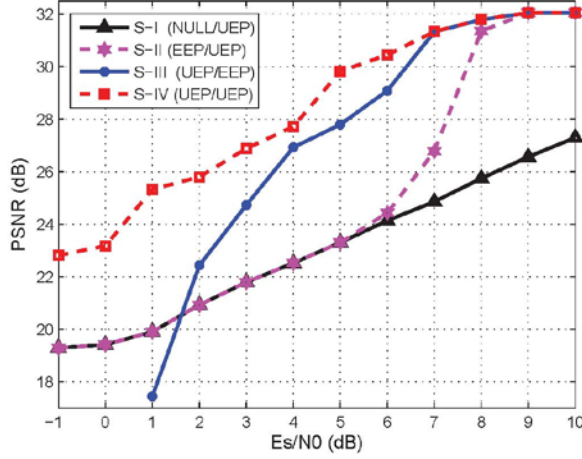
Although our cross-layer FEC parameters were optimized for Bus sequences, the average PSNR performance is similar for the other two test video sequences, i.e., Foreman and Coastguard. As mentioned earlier, both these sequences have different characteristics than the Bus sequence. Thus, we can conclude that the resulting optimum parameters are robust with respect to CMSE.



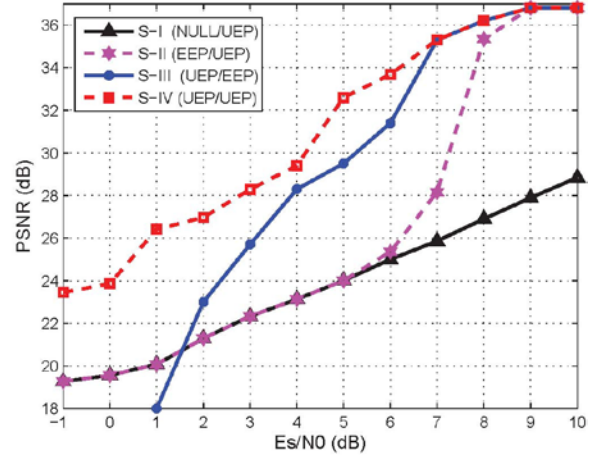
(a)



(b) Bus

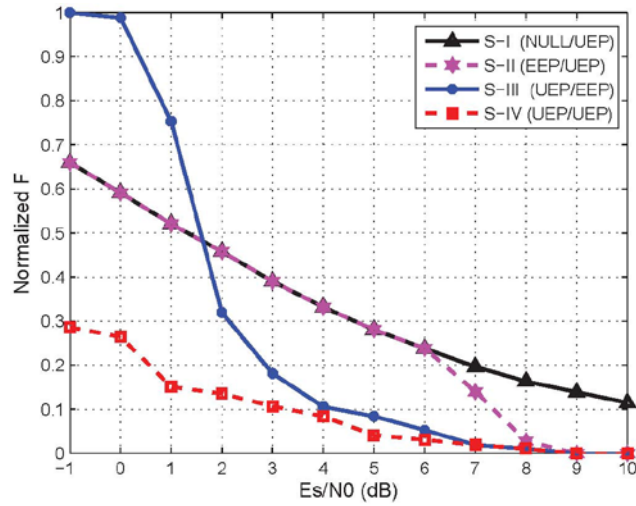


(c) Coastguard

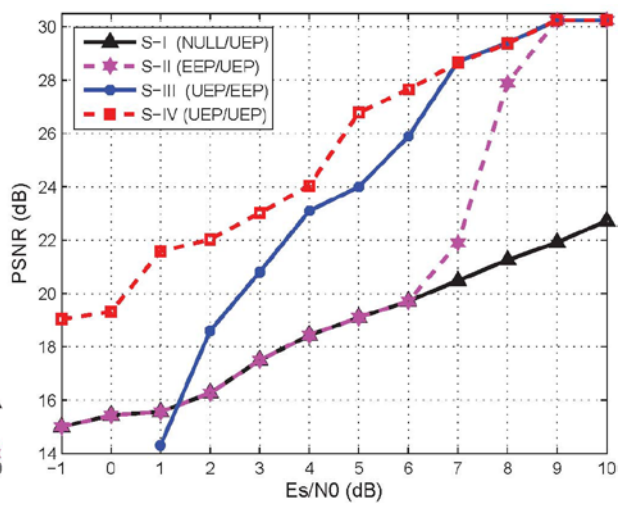


(d) Foreman

Figure 14: Normalized F and average PSNR of test videos for channel SNRs at  $C = 1.4$  Mbps in Rayleigh flat fading channels with  $f_m = 41.7$ ,  $f_c = 900$  MHz, and mobile velocity of 5 km/h.



(a)



(b) Bus

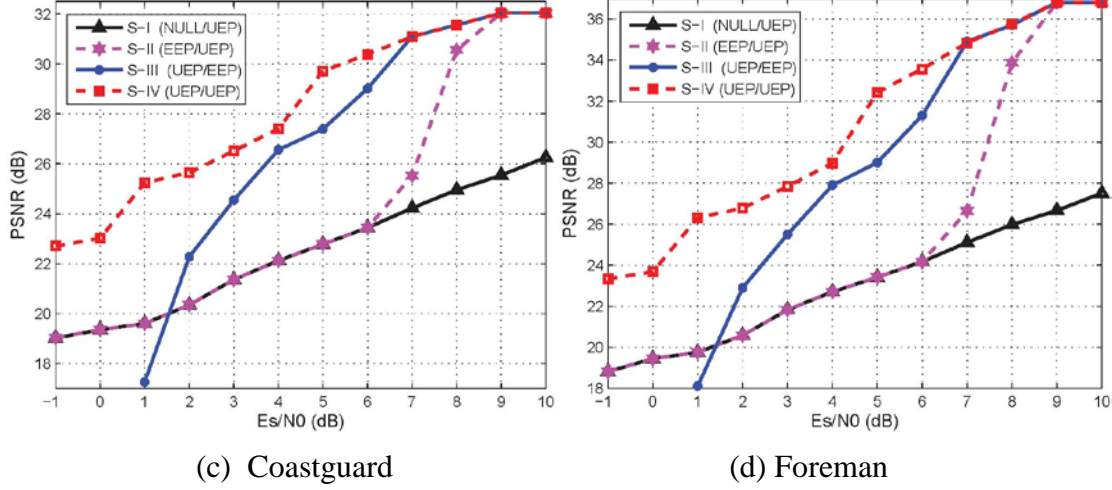


Figure 15: Normalized F and Average PSNR of test videos for various channel SNRs at  $C = 1.4$  Mbps in Rayleigh flat fading channels with  $f_m = 41.7$ ,  $f_c = 900$  MHz, and speed of 50 km/h.

Since Akiyo has considerably different values of  $\overline{\text{CMSE}}_i$ , the proposed S-IV scheme designed by using Bus video's  $\overline{\text{CMSE}}_i$  values may be suboptimal for Akiyo. In order to study the effect of these CMSE variations, we also designed the S-IV scheme by using the  $\overline{\text{CMSE}}_i$  values of Akiyo and compare its performance with its suboptimal version. The optimization results are reported in Table 9. In this table, we also included the suboptimal values of  $F_{\text{sub}}$  and  $\text{PSNR}_{\text{sub}}$ , which were obtained by using the optimized parameters of Bus from Table 8.

In Table 9 (for optimal scheme) and Table 8 (for suboptimal scheme), the Raptor code overhead (i.e.,  $\theta_t$ ) and RCPC code strength ( $R$ ) are the same for both schemes, whereas the



values of Raptor code protection level  $\rho_i$  for each priority class vary slightly (e.g.,  $\rho_1$  is higher for optimal scheme compared to the suboptimal scheme). Similarly, the values of  $VSLR_i$  for higher priority slices (which have the most impact on F and PSNR) are similar in both tables. The maximum PSNR degradation of the suboptimal scheme compared to the optimal scheme is 0.32 dB at the channel SNR of 1.5dB, with only about 0.01 to 0.15 dB PSNR degradation at other channel SNRs. We can, therefore, conclude that the performance of the proposed cross-layer FEC scheme is not very sensitive to the precise values of normalized CMSE.

#### 4.4.3 Performance of Cross-Layer FEC Schemes for Test Videos over Fading Channels

In this section, we evaluate the performance of cross-layer FEC schemes over a Rayleigh flat fading channel with additive white Gaussian noise. We assume the channel to be time-invariant over the duration of one packet and use the instantaneous SNR to characterize the CSI. We use  $\gamma[i]$  to denote the instantaneous SNR of  $i$ th packet. For a Rayleigh flat fading channel, the SNR follows an exponential distribution and can be described by the average SNR [71, 72]. Specifically,  $Pr(SNR < x) = 1 - e^{-\frac{1}{\lambda}x}$  when the average SNR is  $\lambda$ . We can use the past SNR observations from previous transmissions to estimate and update the fading distribution.

In many video streaming applications, Raptor codes are applied on a block of packets of a few video frames or one whole GOP [54, 62, 93]. On the other hand, FEC at the PHY layer is applied at per packet basis using the instantaneous channel SNR. Our cross-layer scheme thus uses two different time scales. It uses the average channel SNR to apply a cross-layer



optimization at a longer timescale (e.g., a two video frame time or one GOP time), and does not assume non-causal channel knowledge. The optimization process for the four FEC schemes is the same as in Section 4.3.2. From the cross-layer optimization, we get the FEC overhead for protecting video data of each priority class at APP layer and a PER constraint which should be achieved at PHY layer by the RCPC code. Then Raptor codes use the optimal allocated overhead for each priority video data to encode the source symbols. For each packet at PHY layer, a suitable RCPC code rate is selected according to the instantaneous SNR and the PER constraint of each priority packet.

We use the Clarke's channel model [94, 95] to simulate BPSK transmission over Rayleigh flat fading channel with Doppler shift in mobile wireless environment. The maximum Doppler frequency is given by  $f_m = \frac{vf_c}{c}$ , where  $f_c$  is the carrier frequency,  $v$  is the mobile velocity, and  $c$  is the speed of light ( $3 \times 10^8$  m/sec). In the experiments, we used  $f_c = 900$  MHz and the propagation paths  $M = 32$ , at two different mobile speeds 5km/h and 50km/h. The experimental results for the cross-layer FEC schemes using one GOP for optimization are shown in Figs. 14 and 15.

Our proposed S-IV scheme achieves a PSNR gain of more than 4 dB compared to the S-I and S-II schemes for  $\frac{E_s}{N_0} \leq 7$  dB. It outperforms S-III scheme by more than 1 dB for  $\frac{E_s}{N_0} \leq 6$  dB.

In Figs. 14 and 15 and Figs. 12 and 13, the performance in a Rayleigh flat fading channel with Doppler shift is worse than in the AWGN channel, especially for scheme S-I which has no Raptor codes at APP. This is because BER decreases linearly in the Rayleigh flat fading channel

and exponentially in the AWGN channel, with increase in the instantaneous SNR [96]. When  $E_s/N_0$  increases, the schemes with UEP Raptor codes at APP (S-III and S-IV schemes) achieve better performance than S-I scheme, which does not use FEC protection (Raptor codes) at APP.

From Figs. 14 and 15, we also observe that the performance degrades more for faster mobile velocity (i.e., larger Doppler shift) because reliable channel estimation becomes difficult when faster variations are introduced in the radio channel.

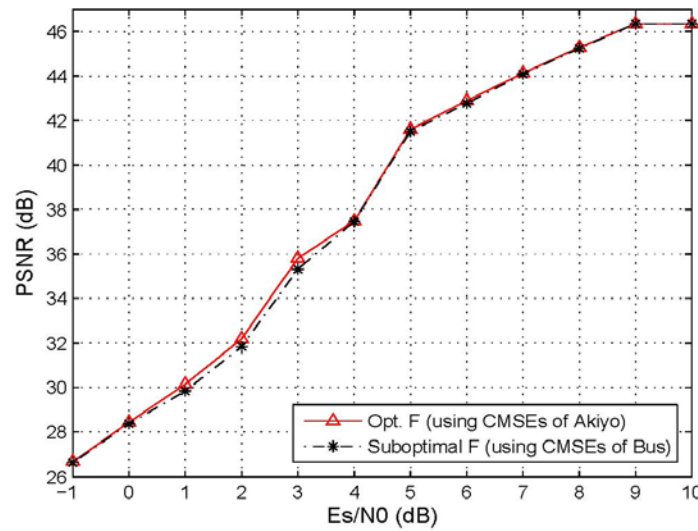


Figure 16: Average PSNR of the optimal and sub-optimal FEC scheme (S-IV) for Akiyo over Rayleigh flat fading channel with  $f_m = 41.7$ ,  $f_c = 900$  MHz at speed of 5 km/h.

Since Akiyo has considerably different values of  $\overline{CMSE}_i$ , the proposed S-IV scheme designed by using Bus video's  $\overline{CMSE}_i$  values may be suboptimal for Akiyo. In order to study the effect of these CMSE variations in fading channel, we also design the S-IV scheme by using the

$\overline{\text{CMSE}}_i$  values of Akiyo and compare its performance with its suboptimal version. The values of  $\text{PSNR}_{opt}$  and  $\text{PSNR}_{sub}$  which were obtained by using the optimized parameters of Akiyo and Bus video, are shown in Figure 16. The maximum PSNR degradation of the suboptimal scheme compared to the optimal scheme is about 0.35 dB at the channel SNRs of 1 dB, 2 dB and 3 dB, with only about 0.01 to 0.15 dB PSNR degradation at other channel SNRs. We can conclude that the performance of the proposed cross-layer FEC scheme is not very sensitive to the precise values of normalized CMSE in fading channel. We had a similar observation for AWGN channels in the previous section.

## 4.5 Conclusions

Previously, the UEP FEC coding at PHY (without any FEC coding at APP) and cross-layer FEC schemes using EEP (or UEP) FEC coding at APP and EEP FEC coding at PHY have been used for video transmission over error-prone wireless channels. However, the joint optimization of cross-layer UEP FEC codes at both the APP and PHY for video transmission has not received due attention. We used the UEP Raptor coding at APP and UEP RCPC coding at PHY for robust H.264 video transmission over error-prone wireless channels. H.264 video slices were prioritized based on their contribution to video quality. We used a probability selection model for Raptor codes to provide UEP for H.264 video slices. Video slices of each priority class were encoded using independent Raptor encoders. We performed the cross-layer optimization to concurrently tune the FEC code parameters at both layers, in order to minimize the video distortion and maximize the peak signal-to-noise ratio (PSNR). We observed that the cross-layer UEP FEC scheme outperformed other FEC schemes that use the UEP coding at APP or PHY, including the

cross-layer FEC schemes, for different channel SNRs and bit rates for AWGN and Rayleigh flat fading channels. Further, we showed that our optimization works well for different H.264 encoded video sequences, which have widely different characteristics.

## **5.0 CROSS-LAYER SCHEDULING SCHEME FOR VIDEO TRANSMISSION OVER WIRELESS NETWORKS**

### **5.1 Introduction**

To provide better video streaming quality over wireless channels, various technologies have been employed such as scalable video coding [3, 97], error resilient coding [4, 98, 99], video transcoding [100, 101], packet scheduling [102, 107, 130], and playout adaptation [103-106]. Scheduling algorithms employed at the transmitter play a key role in determining the performance of wireless systems. Most of the initial work on scheduling schemes focused on maximizing throughput and optimizing system performance for non-real-time and delay-tolerant traffic. For example, opportunistic schedulers for the problem of downlink scheduling were extensively studied in [108, 109], wherein a single transmitter at the base station is shared amongst multiple downlink users. Opportunistic scheduling entails exploiting multiuser diversity inherent in wireless systems due to fluctuating channels. However, such schedulers, being oblivious to packet deadlines, video data bit rate variations, and frame dependencies, perform poorly in the context of delay-sensitive video streaming. Therefore, network-adaptive video streaming techniques proposed in [109-111] have gained significant interest. They try to overcome fluctuations due to wireless link impairments by using controls at various layers of the transmitter and/or receiver.

In a streaming media system, the client usually buffers the video data it has received in a playout buffer and begins playback after a short delay (known as the pre-roll delay) of up to several seconds [112]. Smoothing the video in this manner allows it to be transmitted in a

less bursty fashion and potentially simplifies operations such as resource allocation and improves network utilization [113, 114].

Adaptive streaming techniques are generally classified as either receiver-driven or transmitter-driven [110]. A receiver-driven technique that allows the streaming media client to control the playout rate of the decoder without the involvement of the transmitter was proposed in [115]. Depending on the video and the playout buffer fullness (amount of data in the playout buffer), playout interval variation from 25% up to 50% was considered. Though this reduces the probability of playout buffer underflow and overflow, noticeable artifacts can still occur in the displayed video.

In the transmitter-driven techniques, rate-distortion (R-D) optimized packet scheduling techniques [102, 116, 117] are the state-of-the art. In every transmission opportunity, the rate is optimized for the scheduled media unit (a group of NAL units) to minimize the expected received video distortion by taking into consideration the transmission errors, retransmission delays, the decoding dependencies (frame types), and the channel bit rate constraint. It also includes selecting the media units to discard for a low channel bit rate constraint. The optimization problem is solved for an average channel by using the Lagrangian R-D formulation and is not designed to adapt and exploit the time-varying transmission rates supported by wireless links. Further, though the above schemes could show noticeable benefits by allowing adaptation to wireless link errors and retransmission delays, they require significant modifications in the streaming client and/or the streaming server [118, 119]. Our scheme focuses on solutions to schedule the video stream over a wireless link with time-varying bit rate, which requires insignificant modifications in the streaming server. At the same time, our scheduling solution provides improved video quality at the receiver by considering the relative importance of the frames and their delay bounds.

Transmission rates on the wireless links could vary significantly in every transmission time interval (TTI) due to impairments such as fading, and multi-user channel access characteristics [120, 121]. These changes in transmission rate impact the end-to-end delay of video frames. When the wireless link is slow and cannot support the video bit rate, compressed video frames fill up the post-encoding buffer eventually causing it to overflow and the frames to timeout. Meanwhile, frames are continuously played out at the client, causing the playout buffer to underflow and eventually causing an outage. Buffer underflow occurs when the number of frames in the playout buffer falls below a pre-determined threshold whereas an empty playout buffer results in an outage [105, 122, 123, 144].

Most of the existing transmitter-based scheduling schemes are based on the single layer coding of H.264/AVC [2] and propose modifications to the rate control module of the encoder. The scalable extension of H.264 enables encoding a high-quality video bit stream containing one or more subset bit streams [3]. This makes it attractive to be used in streaming applications. In this section, we propose a transmitter-driven scheduling algorithm which is aware of video packet importance and frame deadlines. It exploits the temporal and SNR scalabilities of a H.264/SVC compressed bit stream, and derives a subset (i.e., scalable) bit stream for transmission over a wireless link with time-varying bit rate. The subset bit stream provides graceful degradation in bad channel conditions. Our scheme uses a sliding-window based flow control at the post-encoding buffer of the streaming server. The flow control determines how many and which particular NAL units, from a window of temporal and quality layers, are to be scheduled for transmission during every TTI. The scheduled NAL units improve the received video quality for the available channel resources. The optimization problem of maximizing the expected received video quality is reduced to

maximizing the product of the normalized CMSE value with the inverse of the time-to-expiry (TTE) value.

## 5.2 Related Work

Kang and Zakhor [124] proposed a packet scheduling algorithm for streaming an MPEG-4 compressed video over wireless channels with dedicated fixed bandwidth, fixed round trip time, and known channel bit error rate. Different deadline thresholds were assigned to video packets based on their importance. The importance of a video packet was determined by its relative position within the GOP and its motion texture context. Packets with the nearest deadline were transmitted first.

A packet selection algorithm for adaptive transmission of smoothed and layered video over a wireless channel was discussed in [125]. Before transmitting a packet from the current video layer, the scheme proposes to compute the minimum success probability of the next higher priority layer among all the remaining frames. Depending on whether this value is greater than a pre-determined heuristic threshold, the packet from the current layer could either be transmitted or discarded. This is done to maintain similar video quality among the transmitted frames. However, the complexity involved in determining the minimum success probability increases as the number of frames increases. Further, a time-varying channel makes it infeasible to compute the success probability for a large number of remaining frames.

Hung et al. [144] proposed a scheduling scheme based on an active and passive playout adaptation in the receiver buffer. The active playout tries to smooth the video playout by slowly varying its rate in order to overcome bad channel conditions. The passive playout kicks in during serious congestion and the smallest possible playout rate is employed at the



receiver buffer. Playout interval variations of up to 50% are considered depending on the video content. However, the playout adaption is still limited in efficiently delivering video packets over a time-varying wireless link and in avoiding playout interruptions. Hence, a deadline-aware packet scheduling scheme is also considered at the transmitter which discards the packets of the frames which have missed their playout deadline. It also uses different numbers of retransmissions for packets belonging to different priority frames and schedules the new packets and the packets to be re-transmitted within channel bit-rate constraints. The scheme does not fully avoid playout buffer outage.

Chen et al. [145] studied an adaptive video scheduling scheme in a Markov decision process (MDP) framework at the transmitter, which requires the knowledge of instantaneous playout buffer status and channel conditions at the receiver. However, the scheduling policy is derived offline and thus is not adaptive to channels with time-varying bit rate. A state space reduction technique is proposed to limit the complexity of the MDP. The scheduling scheme works on a window of frames to be decoded at the receiver. The window size provides a tradeoff between the optimality and complexity of the scheduling scheme.

A priority-based media delivery scheme is discussed by [126] for the pre-buffering and re-buffering in the receiver playout buffer to overcome channel interruptions. The H.264/SVC bit stream is divided into three priorities. The scheduling scheme buffers more high priority data in the playout buffer. This results in pre-buffering the data for a longer playback time compared to the earliest deadline first (EDF) scheme [142]. The scheme has been proposed for both real time protocol (RTP) and hypertext transfer protocol (HTTP) based streaming.

In order to reduce the impact of network bandwidth fluctuation, an adaptive priority ordering algorithm for H.264/SVC bitstreams is proposed in [127]. It arranges the coding

layers (i.e. spatial, temporal, and quality scalability) according to their R-D tradeoff within a GOP so that the transmitted video quality can be preserved over dynamic bandwidth conditions.

Stockhammer et al. [118] derived the required initial buffering delay and the receiver buffer size to avoid playout interruption due to buffer underflow or video packet loss due to buffer overflow while streaming a MPEG-4 encoded variable bit rate video. The conditions were derived for a wireless channel with known packet success probability and for pre-encoded video streams. The problem is solved in the framework of the leaky bucket algorithm in the hypothetical reference decoder or video buffering verifier at the receiver.

Recently, Chen et al. [119] described the strict conditions guiding an x264 encoder to design a bandwidth adaptive rate control for the first time. The rate control in [119] derives an upper and lower bound for the target frame size and the corresponding tightest bounds on the encoder and decoder buffer sizes subject to a strict end-to-end delay over a fast time-varying channel. The encoder then fixes the size of the frame to the average of the upper and lower bounds. The scheme depends on the accuracy of channel estimation at the transmitter. It may cause large variation in bits allocated to different frames, resulting in inconsistent video quality due to the emphasis on a strict end-to-end delay bound over a fast time-varying channel. Further, the rate control does not take into account the importance of the frame and the error propagation it may cause (due to the allocated quantization parameter value) at the receiver. To limit the variation in quality from frame-to-frame, accurate R-D models [128, 129] are required to estimate the target frame size for a targeted quality along with some R-D optimization. This has been ignored in [119] since the emphasis of choice on best rate points may cause large delay jitter.

Dua et al. [130] proposed a channel, deadline, and distortion aware scheduling scheme for streaming H.264/AVC compressed videos to multiple video clients in a wireless communication system. The scheduling problem was studied in a DP framework to minimize the aggregate distortion cost incurred over all receivers. The scheme showed significant PSNR gains over benchmark multi-user scheduling schemes such as the round robin, EDF, and best channel first schemes. Distortion for every video packet in a frame was computed as the MSE contributed by its loss. The packets of a frame were then ordered for scheduling based on their distortion. Scheduling was carried out for the packets of a single head-of-line frame of all users at a time, under the assumption that except for the first I-frame, all the other frames in the video are of equal importance. This ignores the fact that video frames contribute different levels of distortion based on their scene complexity, motion level, and type (I, P, and B).

A MDP framework in [131] was used for cross-layer optimization of scheduling at the post-encoding buffer of a video server, the packet size and scheduling at the MAC layer of the base station, and MAC receiver buffer at the client. The scheme derives a foresighted control policy (i.e., the optimal value function) and the optimal policy (set of actions) by using the value iteration algorithm over a constant bit rate BSC. Due to the large dimensionality of the problem, a strong quantization of the values was considered by the different states. The evaluation of the transition probabilities was done offline using the training video sequences. The authors resort to learning techniques, such as reinforcement learning, in order to estimate the optimal policy and also suggest updating the entries of the transition matrix online at each time instant. However, this is not realistic because the base station needs to simultaneously coordinate with the video server and the wireless video client to differentiate bad policies from good ones in real-time and eliminate them. For streaming

applications, it will degrade the video quality until the learning is finished. Moreover, the reward matrix cannot consider the immediate effect of a selected set of actions on video quality and only has to use the video quality determined at the source. The framework also does not consider the frame delay constraints normally associated with scheduling in streaming applications. The foresighted control policy in [131], maximizing some long-term discounted sum of rewards linked to the video quality, achieved considerable PSNR gains compared to the short-term myopic policy in [132], which maximizes the immediate reward without paying attention to the consequence the current decision may have on future rewards.

The problem of joint adaptive media playout control at the receiver and video motion-aware packet scheduling across the APP and MAC layers at the transmitter was formulated in a MDP framework by [132]. It employed an online reinforcement learning approach with a layered real-time DP algorithm for adaptive video transmission. In addition to the parameters in [131], it also considered the modulation and coding options, provided by the PHY layer in the 802.11a standard, in the set of actions and states. It preemptively varied the playout speed of scenes, based on the motion intensity, to reduce the perceptible effect of playout speed variation. However, the high computational complexity of this scheme makes it unsuitable for real-time delay-sensitive streaming.

Li et al. [105] proposed an MDP-based joint control of packet scheduling at the transmitter and content-aware playout at the receiver, in order to maximize the quality of video streaming over wireless channels. They also proposed a content-aware adaptive playout control (i.e., slowdown) that considers the video content (i.e., motion characteristics in particular). This scheme improved the quality of the received video with only a small amount of playout slowdown which was mainly placed in low-motion scenes where its perceived effect is lower.

## 5.3 Methods, Assumptions and Procedures

### 5.3.1 System Model

#### 5.3.1.1 Scalable Video Coding

The coded video data of H.264/SVC [3] are organized into NAL units, each containing an integer number of bytes. NAL units are classified into video coding layer (VCL) NAL units, which contain coded slices or coded slice data partitions, and non-VCL NAL units, which contain associated additional information. The most important non-VCL NAL units are parameter sets and Supplemental Enhancement Information (SEI). The pre-roll delay and the playout rate are communicated by the streaming server to the client through the SEI [118, 119, 133].

We use hierarchical prediction with a structural encoding/decoding delay of zero [3] as shown in Figure 17(a). The temporal enhancement layers are coded as unidirectionally predicted P-pictures. The darkest colored frames belonging to temporal layer  $T_0$  are encoded as key pictures to limit the distortion propagation within a GOP. Our scalable bitstream contains  $|\bar{\mathbf{T}}|$  temporal layers (where  $|\bar{\mathbf{T}}|$  is the cardinality of the set  $\bar{\mathbf{T}}$ ) with a maximum frame rate of  $f_r$  fps (e.g., 30 fps). The GOP size is then computed as  $2^{(|\bar{\mathbf{T}}|-1)}$ . The figure has  $|\bar{\mathbf{T}}| = 4$ ,  $\bar{\mathbf{T}} = \{T_0, T_1, T_2, T_3\}$  and GOP size of 8.

We consider medium-grain scalability (MGS) for SNR scalability. Our scalable bitstream contains  $|\bar{\mathbf{Q}}|$  quality enhancement layers. Every frame is identified with its index  $f$

which could be  $0, 1, 2, \dots, F$ ;  $F$  being the last frame index of the sequence. A NAL unit belonging to a frame  $f$  and quality enhancement layer  $q \in \bar{Q}$  is identified as  $L_{f,q}$ . The base layer (BL) NAL unit of frame  $f$  is identified as  $L_{f,0}$  with  $q = 0$ . Figure 17(b) shows the motion-compensated prediction dependency between the layers for a GOP size of 4. A vertical arrow denotes a spatial prediction signal from the lower layer being used in the upper layer reconstruction. A non-vertical arrow denotes a lower temporal layer being used in the motion-compensated prediction of a higher temporal layer. Together, they determine the error propagation path spatially and temporally. We use a MGS vector  $[3, 3, 10]$  to divide the  $4 \times 4$  integer transform coefficients into three quality enhancement layers [134, 143].

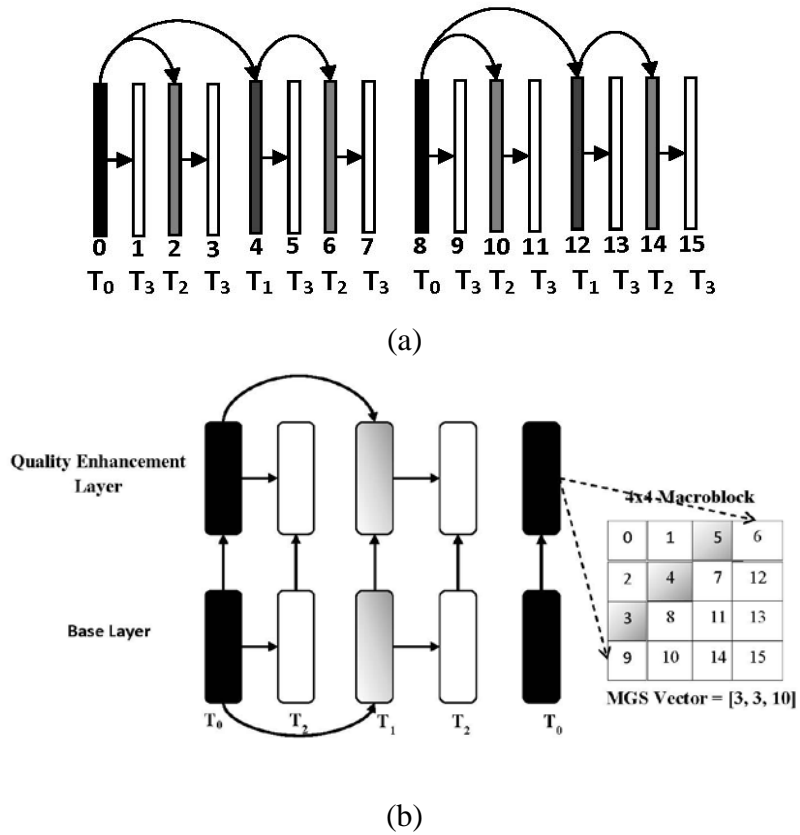


Figure 17: (a) Hierarchical prediction structure, and (b) motion-compensated prediction for MGS layers with key pictures.

Our proposed algorithm uses the CMSE to determine the importance of the VCL NAL units. CMSE values consider the error propagation due to the lost NAL units and are evaluated at the streaming server. CMSE is computed using Equation (1) in Section 3. Figure 18 shows the average R-D characteristic curves for a 480p (720 × 480) video, Table Tennis, compressed using the H.264/SVC codec JSVM 9.8 [134], and using MGS with  $\bar{\mathbf{T}} = \{T_0, T_1, T_2, T_3\}$  and  $\bar{\mathbf{Q}} = \{0, 1, 2, 3\}$ . Every quality layer is represented by a single non-truncatable NAL unit. When the BL of a frame expires, we perform frame copy concealment in the decoder. The y-axis in Figure 18 shows the average distortion (CMSE or IMSE) and the x-axis shows the average bit rate up to a particular temporal layer and quality layer. For example, the four R-D points for temporal frame  $T_0$  correspond to the BL and three quality enhancement layers with corresponding cumulative bit rates of 406, 763, 952, and 1156 Kbps. Similarly, for temporal frame  $T_3$ , the R-D points correspond to the BL and three quality enhancement layers with corresponding cumulative bit rates 593, 1371, 1691, and 2022 Kbps. Maximum video quality is achieved if all the temporal and quality layers are decoded at 2022 Kbps. Similar R-D behavior was observed for other test sequences.

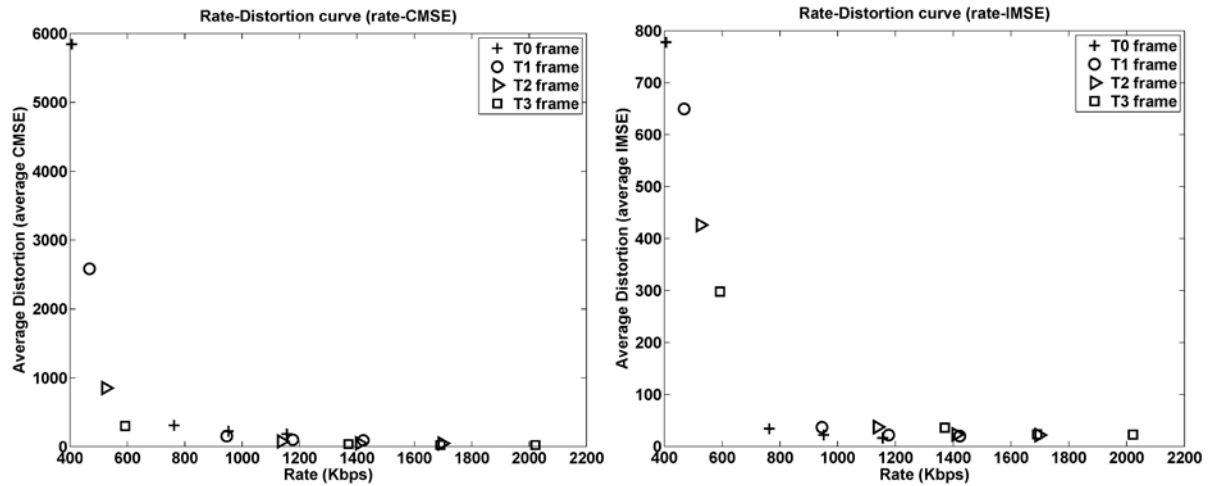


Figure 18: Average R-D characteristic curves in terms of (a) CMSE, and (b) IMSE for different temporal and quality layers.

### 5.3.1.2 Video Streaming System

We consider a wireless video streaming system which consists of a streaming server at the transmitter, a wireless channel, and a streaming client at the receiver as shown in Figure 19. In streaming applications, the video server rather than the encoder decides the rate at which the frames are input into the post-encoder buffer [118, 119]. Hence, the variable bit rate scalable media stream is characterized by a frame duration  $\Delta t$  and a sampling curve  $R_v(t)$ . The sampling curve of the video sequence represents the overall amount of data (measured in bits) delivered into the post-encoder buffer by the video server up to time  $t$ . The sampling curve of the channel indicates the overall amount of video data transmitted up to time  $t$ . The sampling curve is monotonically increasing and has a staircase characteristic.

Figure 20 shows the sampling curve for the 480p Table Tennis video considered in Section



5.3.1.1 at  $\Delta t = \frac{1}{30}$  seconds, i.e., for video frames being delivered into the post-encoder buffer

at 30 fps. The sum average bit rate for all the layers is 2022 Kbps. The non-uniform nature of the jump in the staircase pattern of the video sampling curve is attributed to its bursty nature, i.e. frames with highly fluctuating sizes arrive at a constant interval of  $\Delta t$ . The arrival of a frame belonging to temporal layer  $T_0$  into the post-encoder buffer results in a steeper jump in the Table Tennis sampling curve in Figure 20(b). Figure 20(a) also illustrates two sampling curves for channels supporting different outgoing video bit rates of 1 and 3 Mbps.

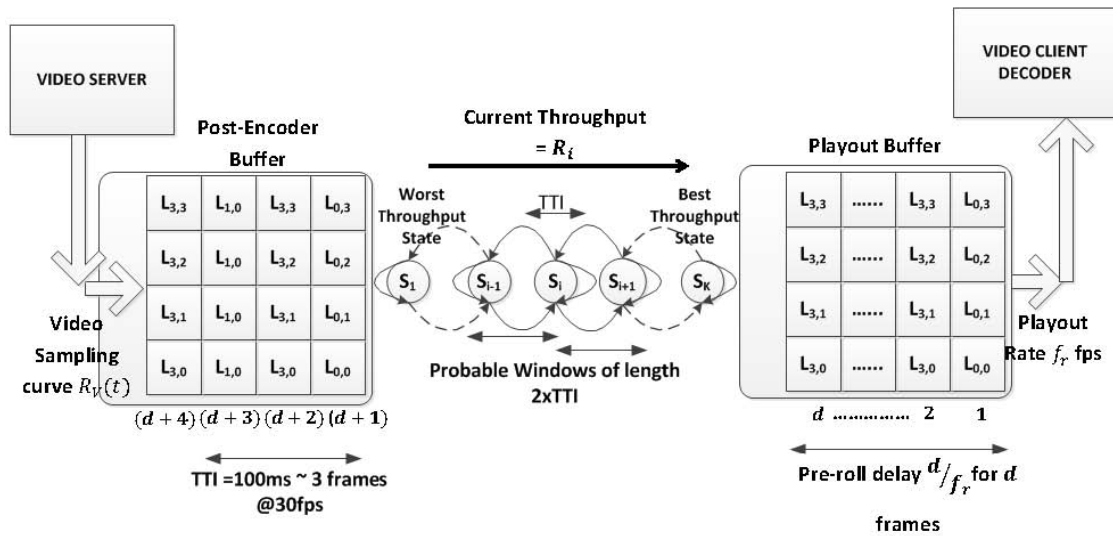


Figure 19: Video streaming system.

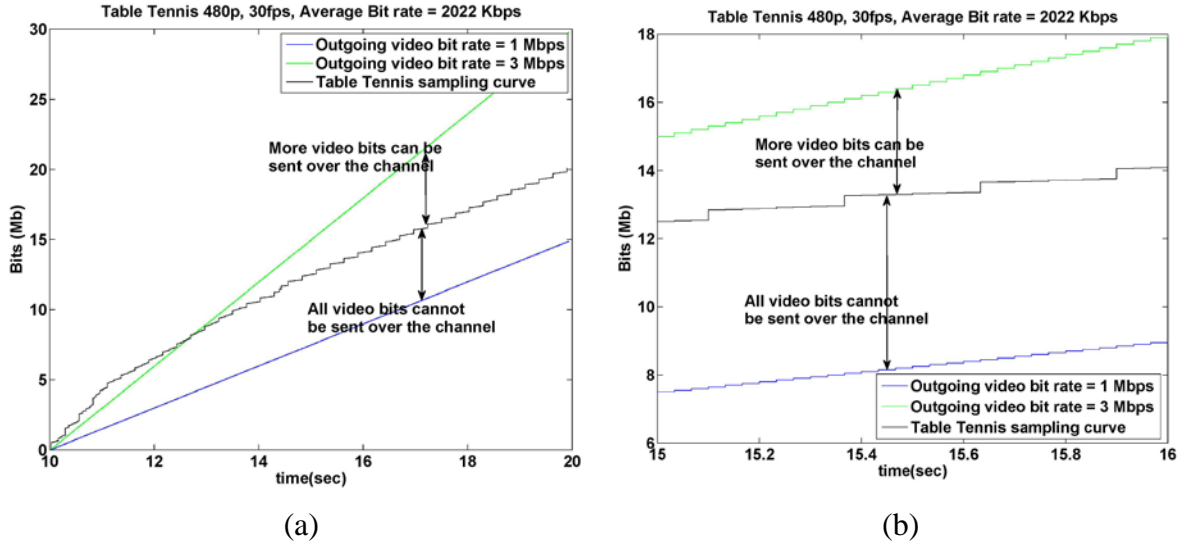


Figure 20: (a) Sampling curve for Table Tennis  $R_v(t)$ , and outgoing video bits supported by the channel, (b) close up of the sampling curve between  $t=15$  sec and  $t=16$  sec.

Frames buffered in the post-encoder buffer have fixed frame deadlines. Frame deadline is the time instant at which the frame is expected by the client for decoding and depends on the pre-roll delay and playout rate allowed at the decoder [111]. All the NAL units of a frame have the same deadline. The pre-roll delay depends on the initial number of frames stored in the playout buffer and the playout rate [14, 109, 110, 115-117]. If  $d$  frames are initially buffered at the receiver, after which it starts decoding and playing them out at a rate of  $f_r$  fps, then the resulting pre-roll delay is  $\frac{d}{f_r}$  seconds. The decoder at the client starts

decoding at time  $t$ . The deadline of a frame  $d+i$  in the post-encoder buffer of the video

server will then be  $t + \frac{d+i}{f_r} = t + \frac{d}{f_r} + \frac{i}{f_r}$ . This is the time at which the client's decoder fetches

the frame  $d + i$  for decoding. If the frame  $d + i$  is not available, then the decoder conceals it using frame copy. Figure 21 illustrates the video streaming timing diagram for a pre-roll delay of  $d = 3$  frames. It shows the times at which the video server begins to transmit the frames in the post-encoder buffer and the times at which they are completely received at the video client. For example, the video server begins transmitting the first frame, belonging to temporal layer  $T_0$ , at time  $t_1$ . The first frame is completely received by the video client at time  $t_2$  (with  $t_2 - t_1$  being the resultant delay). The video server begins the transmission of the second frame, belonging to temporal layer  $T_3$ , at time  $t_2$ . Here, we have ignored the propagation time. The pre-roll condition of  $d = 3$  is satisfied when the third frame, belonging to temporal layer  $T_2$ , is received at the video client at time  $t_4$ . The receiver then starts the decoding process at time  $t_4$ . The video client expects the fourth frame to be available for decoding by  $t_4 + \frac{3}{f_r} + \frac{1}{f_r}$ , which is its frame deadline. The TTE value of a NAL unit is the time duration between the current time and its frame deadline. For example, the current time at which the fourth frame is scheduled in Figure 21 is  $t_4$  and its TTE is equal to  $t_4 + \frac{4}{f_r} - t_4 = \frac{4}{f_r}$ . The TTE of a frame should at least be equal to the time required to transmit

one NAL unit of the frame. If some NAL units of the frame were unable to reach the client within their TTE, then they would expire causing them to be discarded from the post-encoder buffer. The deadlines of the fifth ( $t_4 + \frac{5}{f_r}$ ) and sixth ( $t_4 + \frac{6}{f_r}$ ) frames are also marked on the transmission time axis of the video server and the receive time axis of the video client.

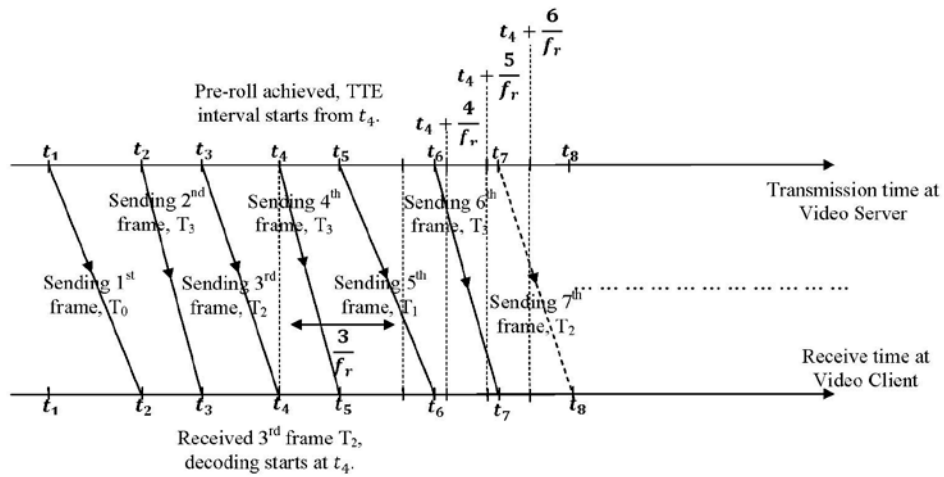


Figure 21: Video streaming timing diagram.

### 5.3.1.3 Wireless Channel

We are interested in capturing the time-varying nature of the wireless channel, whether it is IEEE 802.11, cellular, or home environment, where the available resources are distributed among multiple users and multiple applications. We model the wireless channel as a first-order ergodic Markov chain with  $K$  states, and  $\bar{\mathbf{S}} = \{s_1, s_2, \dots, s_K\}$  denotes its state space [126, 136, 145]. The corresponding video bit rates supported by the states are denoted by  $R_i, i \in \{1, 2, 3, \dots, K\}$ . The channel state supporting the lowest bit rate  $R_1$  is  $s_1$ , and  $s_K$  is the state supporting the highest bit rate  $R_K$ . Let  $p_{i,j}, i, j \in \{1, 2, 3, \dots, K\}$  be the state transition

probability from channel state  $s_i$  to  $s_j$ , and  $\pi_i$  be the steady-state probability of state  $s_i$ . We assume that transitions only happen between adjacent states, i.e.  $p_{i,j} = 0$ , if  $|i - j| > 1$ . The duration of each channel state is equal to one TTI and the constraint on the total number of video bits that can be transmitted in channel state  $s_i$  is  $R_i \times TTI$ . The TTI is considered to be a multiple of frame time, for example  $100\text{ms} \approx 3$  frame time at  $f_r = 30$  fps.

The estimation of the parameters of the Markov model is an important issue. Several studies [136-138] have estimated these parameters from empirical data for some typical environments. Moreover, [120, 121] elaborate on how first-order ergodic Markov chains with different numbers of states can be used to represent a fading channel.

### 5.3.2 Problem Formulation

#### 5.3.2.1 EDF-based Scheme

In existing video transmission systems, packets are transmitted in the same order as they are played out at the receiver. Recent schemes [71, 105, 118, 126, 135, 143-145] have also adopted the EDF [142] motivated scheduling of compressed scalable video for streaming applications. The EDF-based scheme transmits the BL NAL unit followed by the higher SNR layer NAL units of the frame  $f$  with the nearest deadline. The NAL unit of a frame is scheduled only if it can reach the decoder before the frame deadline and this depends on the

supported outgoing video bit rate  $R_i$ . If the BL NAL unit  $L_{f,0}$  expires, the whole frame  $f$  is dropped.

The limitation of the EDF-based scheme becomes evident during persistent bad channel conditions. Even when the channel supports the lowest outgoing video bit rate  $R_1$ , the EDF-based scheme continues to transmit the higher SNR layers of the unexpired frame. This can cause subsequent frames in the post-encoder buffer to be delayed and eventually expire. Though continuous frame losses are concealed using frame copy, they can severely degrade the received video quality. The EDF-based scheme does not consider the importance of different temporal layers and their contribution to distortion.

### 5.3.2.2 CMSE-based Scheme

We try to minimize the expected received video distortion under the constraints of video frame deadlines, and outgoing video bit rates supported by the channel. The CMSE distortion contributed by a NAL unit  $L_{f,q}$  in frame  $f$ , belonging to a temporal layer in  $\bar{\mathbf{T}}$  and spatial layer  $q \in \bar{\mathbf{Q}}$ , is  $D_{f,q}$  and is computed using Equation (1) in Section 3. The size of the NAL unit is  $B_{f,q}$  in bits. Suppose  $d$  frames were allowed to be buffered at the receiver (pre-roll of  $\frac{d}{f_r}$  seconds) after which the receiver started decoding at time  $t'$ . Then at the current

time  $t$ , the TTE of the NAL unit,  $L_{d+i,q}$  in frame  $d + i$ , scheduled to be sent over a channel

with state  $s_l$  is computed as

$$TTE(d + i, t) = t' + \frac{d+i}{f_r} - t \quad (29)$$

If the TTE becomes less than the time required by the NAL unit to reach the decoder, then all the higher SNR layer NAL units in frame  $d + i$  are also discarded along with it. At

the current time  $t$  and channel state  $s_l$ , the TTE of a frame  $f$  must satisfy  $t' + \frac{f}{f_r} - t \geq \frac{B_{f,q}}{R_l}$

for the transmission of its NAL unit  $L_{f,q}$ . Here,  $\frac{B_{f,q}}{R_l}$  is the time required to transmit the NAL

unit  $L_{f,q}$  in the channel state  $s_l$ .

Since the video characteristics and channel rate vary over time, we propose a sliding-window flow control scheme. The algorithm determines which NAL units from a window of  $w(t)$  frames should be scheduled for transmission. The window contains the BL and SNR layer NAL units belonging to unexpired frames which have to be scheduled in the current TTI. When the channel state supports a low outgoing rate then not every NAL unit in the window can be scheduled during the current TTI. Some higher quality layer NAL units which have not expired and were not scheduled in the current TTI remain in the window and get carried over to the next TTI. This increases the number of frames and NAL units to be scheduled.

The flow control optimization is carried out over the window of  $w(t)$  unexpired frames during every TTI to find the set of NAL units  $\bar{\mathbf{N}}$  and their scheduling order which minimizes the expected received video distortion under the constraint of the outgoing rate. The set of all NAL units  $\bar{\mathbf{W}}$  which forms the search space has a size  $w(t) \times |\bar{\mathbf{Q}}|$  and the size of the solution set is  $|\bar{\mathbf{N}}| \leq |\bar{\mathbf{W}}|$ . The search space  $\bar{\mathbf{W}}$  and solution set  $\bar{\mathbf{N}}$  contain 2-tuple elements (frame index, SNR layer id), for example  $(f, n)$ , where  $f$  could be  $0, 1, 2, \dots, F$ , and  $n \in \bar{\mathbf{Q}}$ . The  $j^{th}$  scheduled NAL unit in the solution set  $\bar{\mathbf{N}}$  is accessed as  $\bar{\mathbf{N}}_j(1)$ , and  $\bar{\mathbf{N}}_j(2)$ . To minimize the expected received video distortion in the current TTI where the channel state is  $s_t$ , we must find and schedule the set of NAL units  $\bar{\mathbf{N}}$  which maximizes the objective function formulated as

$$\begin{aligned}
& \max_{\{\bar{\mathbf{N}} \in \bar{\mathbf{W}}\}} \left( \sum_{j=1}^{|\bar{\mathbf{N}}|} D_{\bar{\mathbf{N}}_j(1), \bar{\mathbf{N}}_j(2)} \right) \\
& s. t. \quad (C1): TTE(\bar{\mathbf{N}}_j(1), t) > \frac{B_{\bar{\mathbf{N}}_j(1), \bar{\mathbf{N}}_j(2)}}{R_t}, \quad \forall j \\
& \quad (C2): \sum_{j=1}^{|\bar{\mathbf{N}}|} B_{\bar{\mathbf{N}}_j(1), \bar{\mathbf{N}}_j(2)} \leq (R_t \times TTI).
\end{aligned} \tag{30}$$

The above objective function assumes that a new TTI starts at the current time  $t$ . The first constraint in Equation (30) ensures that only those NAL units are scheduled which can make it to the destination without expiring. The second constraint requires that all the NAL units scheduled in the current TTI must be supported by the rate  $R_t$  for the current channel



state  $s_t$ . The unexpired NAL units belonging to the set  $\{\bar{W} - \bar{N}\}$  remain in contention to be scheduled in the next TTI.

The scheduling problem in Equation (30) is a 0-1 knapsack problem [139, 140] in which each NAL unit is unique as an item, therefore making the number of such copies being selected either 0 or 1. For every item, which is a NAL unit  $L_{\bar{N}_j(1), \bar{N}_j(2)}$ , its distortion  $D_{\bar{N}_j(1), \bar{N}_j(2)}$  represents the item value and its size  $B_{\bar{N}_j(1), \bar{N}_j(2)}$  represents the item weight. The maximum weight supported by the channel is  $R_t \times TTI$ , which represents the number of bits that can be scheduled during that TTI. Each item also has an additional parameter in terms of its TTE value which must satisfy a lower bound (i.e., constraint (C1) in Equation (30)) in order to be in contention to be selected. It is not feasible to solve the formulation in Equation (30) directly by exhaustive search [139, 140].

**Solution using Dynamic Programming:** We solve the optimization problem in Equation (30) using a DP approach which runs in polynomial time (in the number of NAL units scheduled and transmitted). In each iteration, we select one of the unexpired NAL units from the window of  $w(t)$  frames to be scheduled such that the cumulative sum of the CMSE values of the scheduled NAL units is maximum. Basically, the unexpired NAL units which are contending to be scheduled are ranked based on their CMSE contribution and the one with the highest rank is transmitted in each iteration. Further, when more than one NAL unit have the same CMSE, they are ranked depending on the temporal and SNR layers to which

they belong. This scheme gives a higher priority to the NAL units belonging to the lower temporal and SNR layers in the window. These NAL units are generally larger in size and usually contribute high CMSE distortion due to error propagation.

Note that the NAL unit selected in each iteration is a unique solution due to the implicit constraint that the higher SNR layers of a frame cannot contend for selection if its lower layer has not yet been scheduled. Therefore, there will be a maximum of  $w(t)$  NAL units contending in each iteration out of which one NAL unit is selected. Suppose  $\theta$  NAL units from index  $j = k - \theta$  to  $j = k - 1$  have already been scheduled from the search space  $\bar{W}$  in the current TTI (i.e., the size of the current solution set  $|\bar{N}|$  is  $\theta$ ). Further, say the NAL units contending for the current scheduling spot (index  $j = k$ ) belong to a subset  $\tilde{W} \in \bar{W}$ , whose size is  $w(t)$ . Then the  $k^{th}$  NAL unit is selected recursively as,

$$\begin{aligned} \max_{\{\bar{N} \in \bar{W}\}} \left( \sum_{j=k-\theta}^k D_{\bar{N}_j(1), \bar{N}_j(2)} \right) &= \sum_{j=k-\theta}^{k-1} D_{\bar{N}_j(1), \bar{N}_j(2)} \\ &+ \arg \max \left\{ D_{\tilde{W}_1(1), \tilde{W}_1(2)}, \dots, D_{\tilde{W}_{|\tilde{W}|}(1), \tilde{W}_{|\tilde{W}|}(2)} \right\} \end{aligned} \quad (31)$$

Equation (31) implies that the next step of the optimization process is independent of its past steps, thus forming the foundations of the DP solution. The computational complexity is greatly decreased to  $O(|\bar{N}|)$ , depending only on the total number of NAL units scheduled in the TTI. The NAL unit selected in every recursion of Equation (31) is immediately

transmitted. This is a significant improvement over the exponential computational complexity of the exhaustive search algorithm.

### 5.3.2.3 Proposed Scheme

The above CMSE-based scheme does not consider the size (in bits) and the TTE values to rank the contending NAL units. Many NAL units with a large CMSE value also have a large size. Scheduling such a NAL unit may cause more delay to the transmission of subsequent NAL units in the window. We propose a scheduling scheme which considers the importance of NAL units in terms of (a) the CMSE distortion contributed to the received video quality, (b) the size of the NAL unit in bits, and (c) the TTE of the NAL unit in seconds.

We define a new parameter  $\mathcal{V}_{L_{\tilde{\mathbf{W}}_j(1)}, \tilde{\mathbf{W}}_j(2)}(t)$  to rank every contending NAL unit  $L_{\tilde{\mathbf{W}}_j(1)}, \tilde{\mathbf{W}}_j(2)$  in the window  $\tilde{\mathbf{W}}$  by combining these three parameters. At current time  $t$ , the TTE of  $L_{\tilde{\mathbf{W}}_j(1)}, \tilde{\mathbf{W}}_j(2)$  will be  $TTE(\tilde{\mathbf{W}}_j(1), t)$  and it must satisfy  $TTE(\tilde{\mathbf{W}}_j(1), t) > \frac{B_{\tilde{\mathbf{W}}_j(1)}, \tilde{\mathbf{W}}_j(2)}}{R_l}$ .

$\mathcal{V}_{L_{\tilde{\mathbf{W}}_j(1)}, \tilde{\mathbf{W}}_j(2)}(t)$  is computed as

$$\mathcal{V}_{L_{\tilde{\mathbf{W}}_j(1)}, \tilde{\mathbf{W}}_j(2)}(t) = \frac{D_{\tilde{\mathbf{W}}_j(1)}, \tilde{\mathbf{W}}_j(2)}}{B_{\tilde{\mathbf{W}}_j(1)}, \tilde{\mathbf{W}}_j(2)} \times \left( TTE(\tilde{\mathbf{W}}_j(1), t) - \left( \frac{B_{\tilde{\mathbf{W}}_j(1)}, \tilde{\mathbf{W}}_j(2)}}{R_l} \right) \right) \quad (32)$$

In  $\mathcal{V}_{L_{\tilde{w}_j(1), \tilde{w}_j(2)}}(t)$ , the CMSE of the NAL unit divided by its size is its normalized CMSE value while its TTE is updated continuously as time  $t$  progresses. During every iteration of the DP solution, we simply transmit the NAL unit with the maximum  $\mathcal{V}_{L_{\tilde{w}_j(1), \tilde{w}_j(2)}}(t)$  instead of transmitting the NAL unit with the maximum CMSE (shown in Equation (31)).

Figure 22 illustrates a sample of the iterations of our proposed DP solution in a TTI. In Figure 22(a), frames  $j$  to  $j + 4$  constitute the window of frames  $w(t)$  which are considered for transmission during the current TTI. The frame TTE value increases from frame  $j$  to frame  $j + 5$ . The empty spaces in frames  $j$  and  $j + 1$  indicate the NAL units that were transmitted in the previous TTI. The leftover NAL units in frames  $j$  and  $j + 1$  have been carried over to the current TTI. The new frames in the current window are  $j + 2$ ,  $j + 3$ , and  $j + 4$  and the window size is  $w(t) = 5$  frames. Figure 22(b) shows the window after four iterations. The additional empty spaces in the figure indicate the NAL units that have already been transmitted in the current TTI. Figure 22(c) shows the iterations corresponding to the NAL units transmitted in the current TTI. In each iteration, the lowest available SNR layer NAL unit of each frame in the window contend with one another for a scheduling spot. Among the contending NAL units, the one contributing the maximum parameter value (Equation (31)) is chosen for transmission. For example, the BL ( $q = 0$ ) NAL unit of the

frame  $j + 2$ ,  $L_{j+2,0}$ , gets selected for transmission in iteration 1. In iteration 2, the first SNR layer NAL unit of frame  $j + 2$ ,  $L_{j+2,1}$ , comes into contention for a scheduling spot. However, the BL NAL unit of frame  $j + 4$  gets transmitted in iteration 2, and the first SNR layer NAL unit of frame  $j + 2$  is transmitted in iteration 3. During this period, frame  $j$  expired. The window size then decreases to only 4 frames, i.e.  $j + 1$  to  $j + 4$  as shown in Figure 22(b). In iteration 4, only four NAL units now contend against each other for a scheduling spot and the BL of frame  $j + 3$  is scheduled to be transmitted.

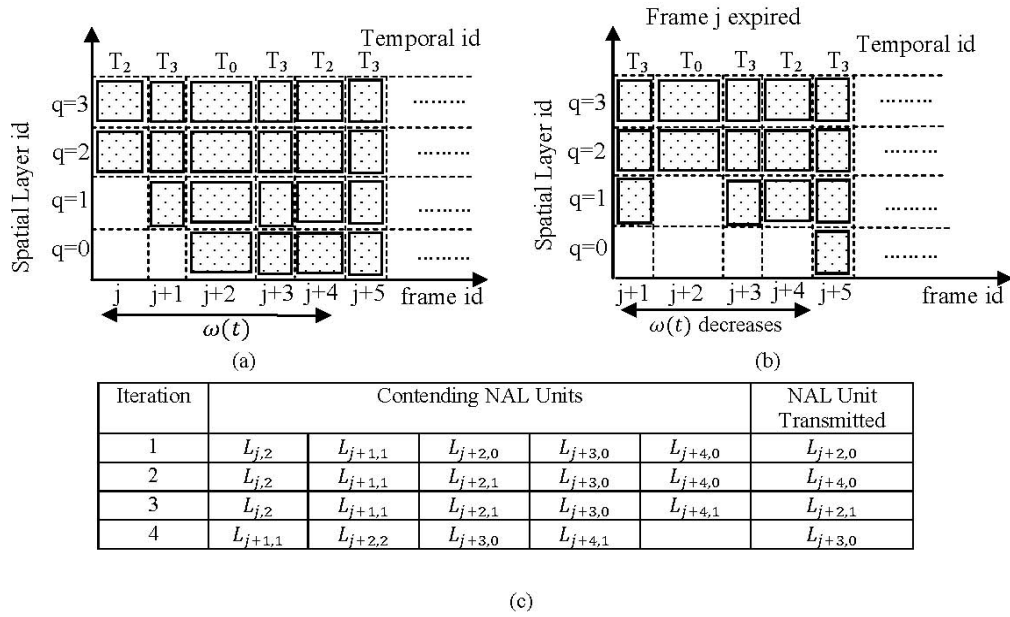


Figure 22: Sample iterations of our proposed dynamic programming algorithm over a window of frames at the video server.

## 5.4 Results and Discussion

We study the effect of the scheduled NAL units on the received video quality through simulations and compare the performance of our proposed approach to (i) the EDF-based scheduling scheme [142], which has also been used recently in [71, 143-145], and (ii) the CMSE-based scheme where the NAL units in the sliding-window are scheduled based only on their CMSE contribution. In the past, frame importance and motion-texture have been used to schedule the frames in non-scalable video streaming [124]. Recently, [22] also used CMSE to prioritize non-scalable NAL units within a GOP and schedule them in the decreasing order of priority. The CMSE-based scheme on scalable video is similar to [22, 124]. Our proposed algorithm trades off the importance of the NAL units with their deadlines and determines the appropriate transmission order for the NAL units in the sliding-window. It significantly reduces whole frame losses and improves received video quality.

### 5.4.1 Simulation Setup

This section evaluates the performance of the EDF-based, CMSE-based, and our proposed scheduling schemes. Two 480p ( $720 \times 480$ ) resolution video sequences, Table Tennis and Stefan, are used in our experiments. They are encoded using H.264/SVC JSVM 9.8 reference software [134] at a frame rate of 30 fps, for a GOP length of 8 frames, using hierarchical prediction with a structural encoding/decoding delay of zero as shown in Figure 17. A GOP size of 8 gives four temporal layers,  $\bar{\mathbf{T}} = \{T_0, T_1, T_2, T_3\}$ . MGS is enabled to achieve a fine level of SNR quality and the integer transform coefficients of every  $4 \times 4$

transform block are split into three additional layers by using the MGS vector [3,3,10] suggested in [134, 143]. Hence, we get four SNR quality layers  $\bar{\mathbf{Q}} = \{0,1,2,3\}$ . Decoding all the temporal and quality layers in Table Tennis and Stefan results in PSNR values of 35.2 dB and 34.8 dB, respectively. Tables 10(a) and 10(b) show the cumulative bit rates of the sub-streams in Table Tennis and Stefan. For example, the bit rate of the BL in temporal layer  $T_1$  in Table Tennis is 468 Kbps, and it includes the BL of temporal layer  $T_0$  from which it is temporally predicted. Similarly, the bit rate for the first quality enhancement layer of temporal layer  $T_2$  in Table Tennis is 1138 Kbps which includes its own BL as well as the BL and first quality enhancement layers of temporal layers  $T_0$  and  $T_1$ . The video playout rate at the receiver is fixed at 30 fps. Four different pre-roll delay values of 0.1, 0.2, 0.3, and 0.4 seconds are considered corresponding to 3, 6, 9, and 12 frames allowed to be initially buffered at the receiver before starting decoding. Each temporal layer has four NAL units corresponding to the four quality enhancement layers. The NAL unit sizes vary depending on the temporal and quality layers and the video content. Generally, the NAL unit size decreases from temporal layer  $T_0$  to  $T_1$ ,  $T_2$ , and  $T_3$ . Tables 11 and 12 show the average NAL unit sizes and average CMSE values.

Table 10: Bit rates (Kbps) of sub-streams of (a) Table Tennis and (b) Stefan

(a)

$(q \in \bar{\mathbf{Q}} \ t \in \bar{\mathbf{T}})$	$T_0$	$T_1$	$T_2$	$T_3$
q = 1	406	468	525	593
q = 2	763	946	1138	1371
q = 3	952	1177	1410	1691
q = 4	1156	1422	1697	2022

(b)

$(q \in \bar{\mathbf{Q}} \ t \in \bar{\mathbf{T}})$	$T_0$	$T_1$	$T_2$	$T_3$
q = 1	506	697	901	1069
q = 2	893	1324	1833	2354
q = 3	1081	1594	2185	2785
q = 4	1199	1743	2360	2995

The wireless channel is modeled as an ergodic Markov chain with three states good,

medium, and bad. The state transition probability matrix  $\bar{\mathbf{P}} = [p_{i,j}] = \begin{bmatrix} 5/6 & 1/6 & 0 \\ 1/6 & 1/2 & 2/6 \\ 0 & 1/3 & 2/3 \end{bmatrix}$

with  $i, j \in \{1, 2, 3\}$ ,  $s_1$  being the bad channel state, and  $s_3$  being the good channel state [126,

135, 143, 145]. We assume that transitions only happen between adjacent states, i.e.,  $p_{i,j} = 0$ ,

if  $|i - j| > 1$ . The state probability vector  $\bar{\mathbf{p}}[l] = [p_{s_1}[l] \ p_{s_2}[l] \ p_{s_3}[l]]^T$  at index  $l$  is

computed using the recursive Chapman-Kolmogorov equation as  $\bar{\mathbf{p}}^T[l] = \bar{\mathbf{p}}^T[l - 1] \times \bar{\mathbf{P}}$ . The



steady-state vector  $\bar{\pi} = \lim_{l \rightarrow +\infty} \bar{\mathbf{p}}[l]$  is computed by solving the system of equations

$\bar{\pi}^T = \bar{\pi}^T \times \bar{\mathbf{P}}$  [141]. The steady-state probabilities of the three channel states are all 1/3.

The frames are read into the post-encoder buffer at 30 fps. The TTI value of the channel is set to 100 ms which is equal to a window of approximately 3 frames. The supported outgoing video bit rates,  $R_i$ , corresponding to the good, medium, and bad channel states for Stefan are 3000, 2100, and 1200 Kbps, and for Table Tennis are 2025, 1400, and 800 Kbps. Monte-Carlo simulations were performed for 120 random channel realizations. Each channel realization contains multiple channel states of TTI duration. To verify that 120 random channel realizations are a sufficient number, we generated two additional sets of 120 realizations each and verified that the average output results were within 0.0005%. The EDF-based, CMSE-based, and our proposed scheduling schemes are depicted in the figures as 'EDF', 'CMSE', and 'Prop.'.

Table 11: (a) Average NAL unit sizes (bytes) and (b) average CMSE values of Table Tennis

(a)				
$(q \in \bar{\mathbf{Q}} \ t \in \bar{\mathbf{T}})$	$T_0$	$T_1$	$T_2$	$T_3$
q = 1	13522	1942	892	568
q = 2	11907	3933	2221	1374
q = 3	6302	1336	673	405
q = 4	6791	1328	676	368

(b)				
$(q \in \bar{\mathbf{Q}} \ t \in \bar{\mathbf{T}})$	$T_0$	$T_1$	$T_2$	$T_3$
q = 1	5845	2582	851	298
q = 2	308	152	75	36
q = 3	230	98	48	23
q = 4	190	91	46	23

Table 12: (a) Average NAL unit sizes (bytes) and (b) average CMSE values of Stefan.

(a)

$(q \in \bar{\mathbf{Q}} \ t \in \bar{\mathbf{T}})$	$T_0$	$T_1$	$T_2$	$T_3$
$q = 1$	16882	6222	3354	1401
$q = 2$	12903	7955	5027	2941
$q = 3$	6265	2716	1342	659
$q = 4$	3942	967	422	299

(b)

$(q \in \bar{\mathbf{Q}} \ t \in \bar{\mathbf{T}})$	$T_0$	$T_1$	$T_2$	$T_3$
$q = 1$	10591	5626	2585	1022
$q = 2$	288	175	95	44
$q = 3$	211	106	52	26
$q = 4$	177	92	48	25

#### 5.4.2 Evaluation of Average Goodput and Percentage of Expired Whole Frames

We first compute the goodput (defined as the ratio of the total video bits received to the total video bits in the sequence) for all the scheduling schemes. Figures 23(a) and 23(b) show the average goodput evaluated over 120 different channel realizations for Table Tennis and Stefan. The average goodput values for the EDF-based, CMSE-based, and proposed schemes differ only within 1.2%. Also the average goodput increases with pre-roll delay because more frames are allowed to be buffered at the receiver which increases the frame deadlines and TTE values of the NAL units in the post-encoder buffer.

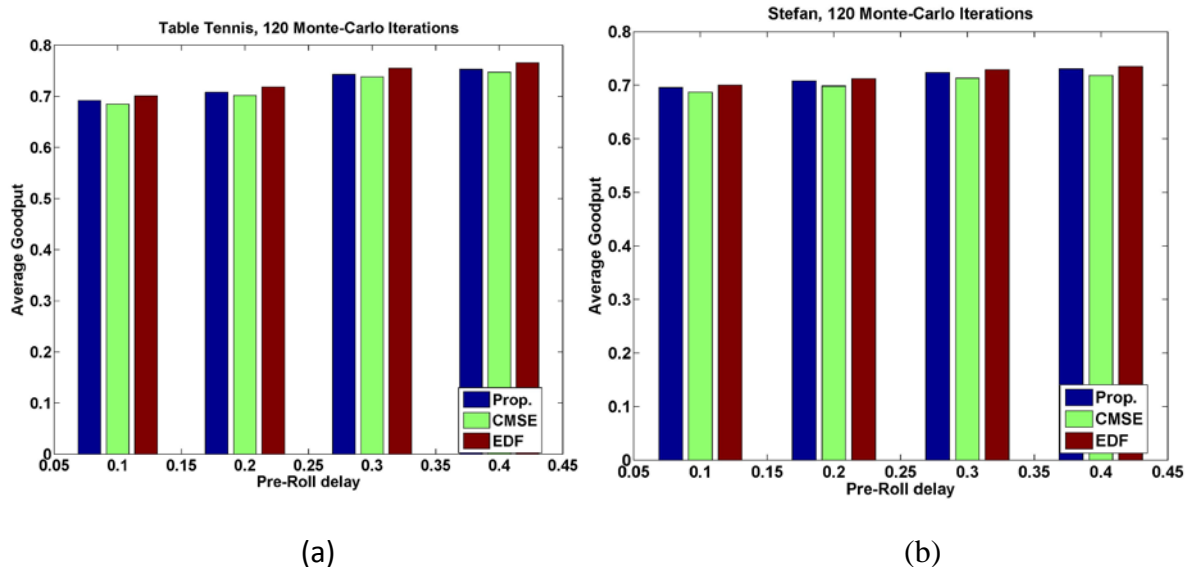


Figure 23: Average goodput of the EDF-based, CMSE-based, and proposed scheduling schemes.

A frame is completely lost if its BL NAL unit expires. Figures 24(a) and 24(b) show the percentage of expired whole frames averaged over 120 channel realizations, for Table Tennis and Stefan in the EDF-based, CMSE-based, and proposed schemes at different pre-roll delays. The expired whole frames are discarded from the post-encoder buffer and concealed at the decoder by using frame copy. The CMSE-based scheme sends the most important NAL units belonging to the lower temporal and SNR layers and hence incurs a lower percentage of expired whole frames compared to the EDF-based scheme. However, it ignores the frame deadlines causing frames in higher temporal layers (e.g.,  $T_3$ ) to expire. The proposed scheme achieves a very low percentage of whole frame losses because it considers the TTE value, CMSE contribution, and the sizes of the NAL units in the frame window. As the pre-roll delay increases, the percentage of expired whole frames decreases in all three schemes. As discussed earlier, a higher pre-roll delay results in higher frame deadlines and

NAL unit TTEs. This reduces the number of NAL units that expire, due to the increased transmission delays during bad channel conditions.

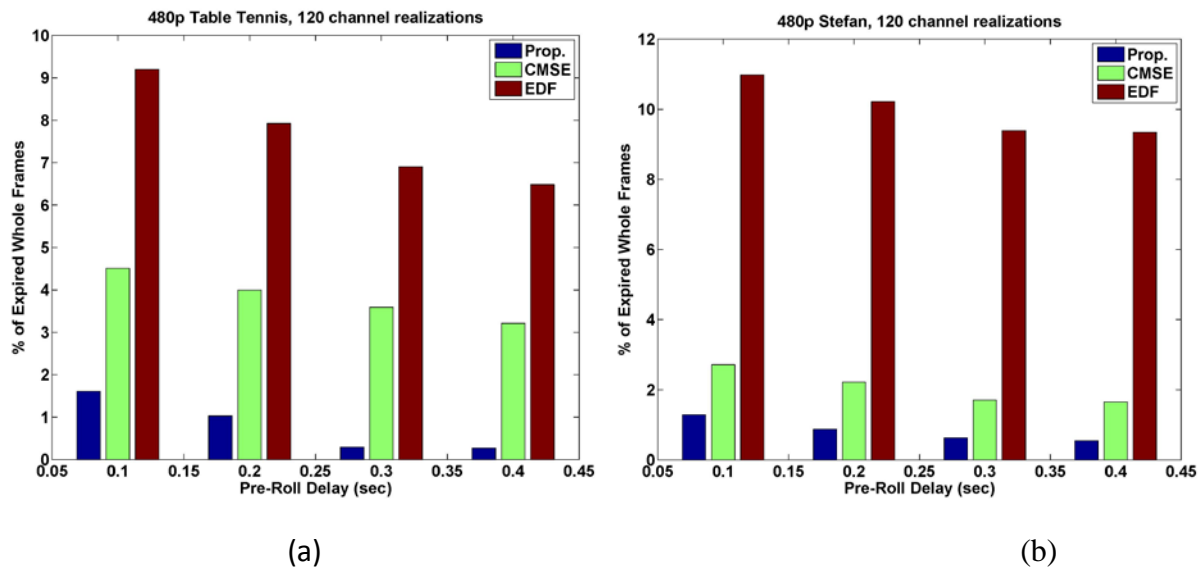
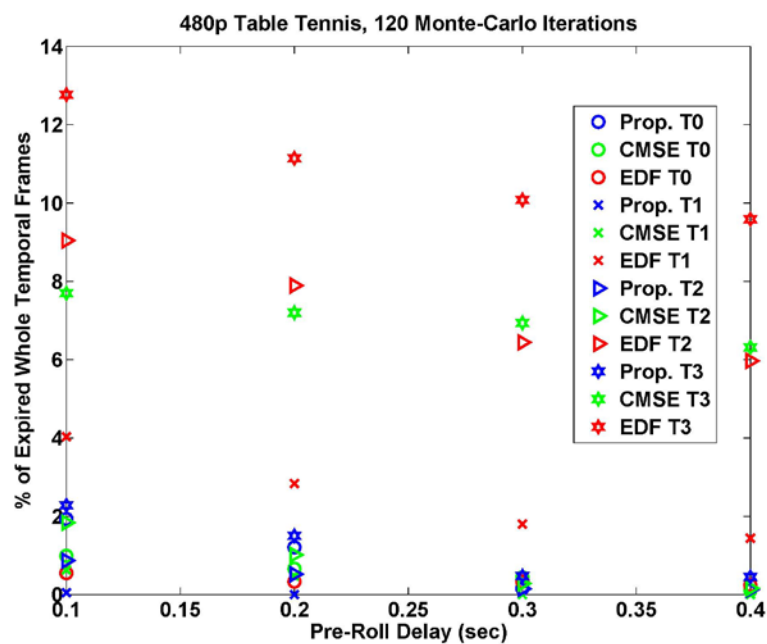
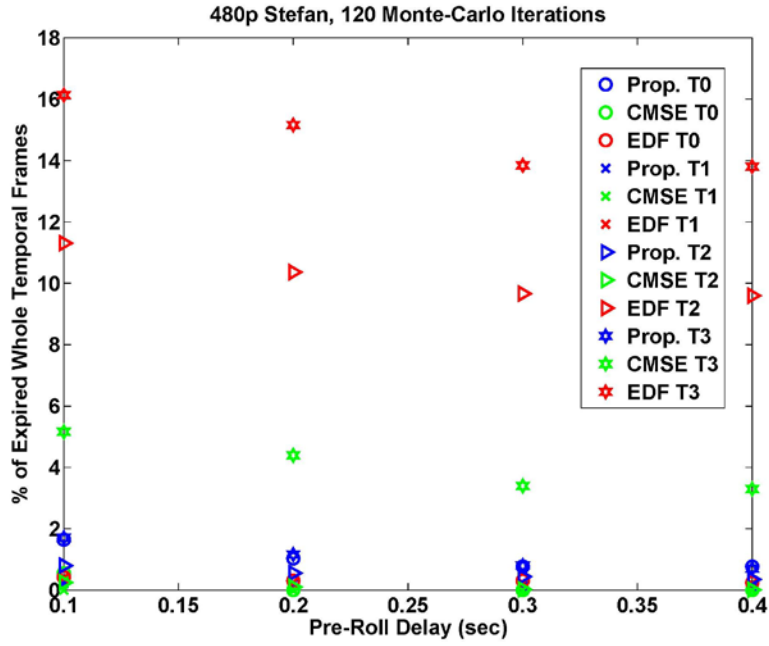


Figure 24: Percentage of expired whole frames in EDF-based, CMSE-based, and proposed schemes over 120 random channel realizations.



(a)



(b)

Figure 25: Percentage of expired whole frames in different temporal layers of EDF-based, CMSE-based, and proposed schemes over 120 random channel realizations.

Figures 25(a) and 25(b) show the percentage of expired whole frames from different temporal layers in Table Tennis and Stefan, computed as a ratio of the number of frames in a temporal layer whose BL NAL unit has expired to the total number of frames in that layer, averaged over 120 random channel realizations. The EDF-based scheme discards a significantly higher percentage of frames belonging to the higher temporal layers  $T_2$  and  $T_3$  as compared to the CMSE-based and proposed schemes. Since the EDF-based scheme considers only the TTE values of the NAL units during scheduling, transmission of the significantly larger frames belonging to  $T_0$  and  $T_1$  cause the smaller sized frames belonging to  $T_2$  and  $T_3$  to expire. The CMSE-based scheme ignores the frame deadlines and only considers the CMSE values of the NAL units. From Tables 11 and 12, we observe that this scheme transmits the larger BL NAL units of lower temporal layers in the window causing more NAL units belonging to higher temporal and SNR layers to expire. Though the

proposed scheme considerably reduces the total number of expired whole frames, it incurs a slightly higher percentage of expired frames from  $T_0$  as compared to the CMSE-based and EDF-based schemes. Though the CMSE distortion contributed by a NAL unit in  $T_0$  is large, sometimes its size is also large causing its normalized CMSE to become smaller than other contending NAL units. This causes it to lose out to NAL units from higher temporal layers while contending for a scheduling spot. Table 13 shows the average normalized CMSE values for Table Tennis and Stefan derived from Tables 11 and 12, respectively.

Table 13: Average normalized CMSE values of (a) Table Tennis and (b) Stefan.

(a)				
$(q \in \bar{\mathbf{Q}} \ t \in \bar{\mathbf{T}})$	$T_0$	$T_1$	$T_2$	$T_3$
$q = 1$	0.43	1.33	0.95	0.5
$q = 2$	0.03	0.04	0.03	0.03
$q = 3$	0.04	0.07	0.07	0.06
$q = 4$	0.03	0.07	0.07	0.06

(b)				
$(q \in \bar{\mathbf{Q}} \ t \in \bar{\mathbf{T}})$	$T_0$	$T_1$	$T_2$	$T_3$
$q = 1$	0.63	0.91	0.77	0.73
$q = 2$	0.02	0.02	0.02	0.02
$q = 3$	0.03	0.04	0.04	0.04
$q = 4$	0.05	0.1	0.1	0.1

Next, we look at how the expired NAL units are distributed among the different temporal and SNR layers.

#### 5.4.3 Evaluation of Expired NAL Units

Figures 26(a) and 26(b) illustrate the percentage of NAL units expired in Table Tennis and Stefan, averaged over 120 random channel realizations. The percentage of expired NAL units decreases with increasing pre-roll delay in the three schemes. At every pre-roll

delay, more NAL units are discarded in the proposed scheme than in the EDF-based scheme, for both the sequences. The CMSE-based scheme has the highest percentage of expired NAL units among the three schemes. However, the goodput is almost the same for the three schemes as shown in Figure 23. In fact, more higher SNR layer NAL units expire in CMSE-based and our proposed schemes, which is discussed in the next paragraph. As shown in Tables 11 and 12, these higher SNR layer NAL units are much smaller in size than the BL NAL units. Since both the CMSE-based scheme and our proposed scheme schedule the larger BL NAL units from the frame window more often, the NAL units belonging to higher SNR layers expire.

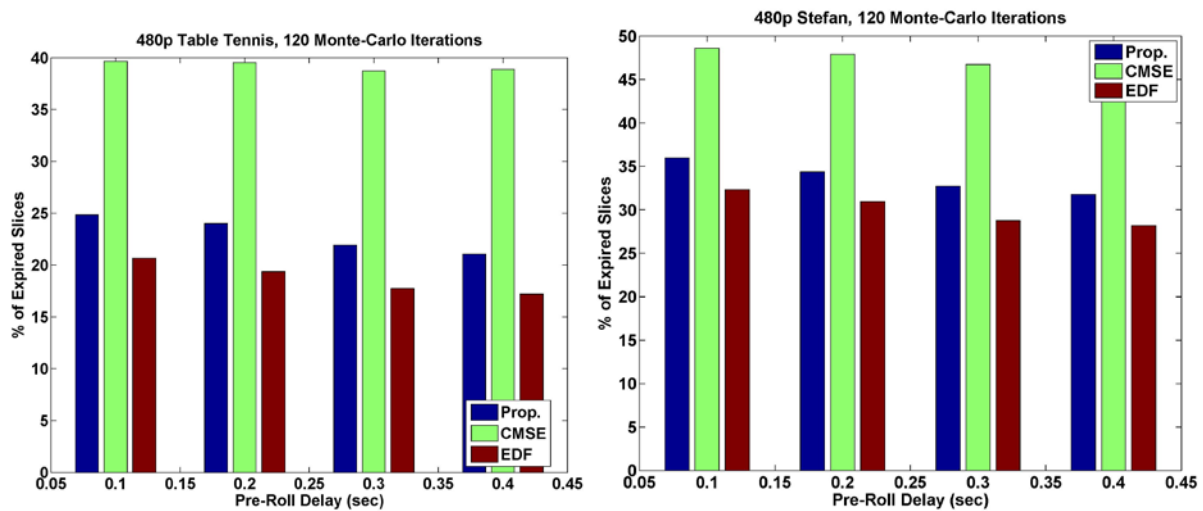
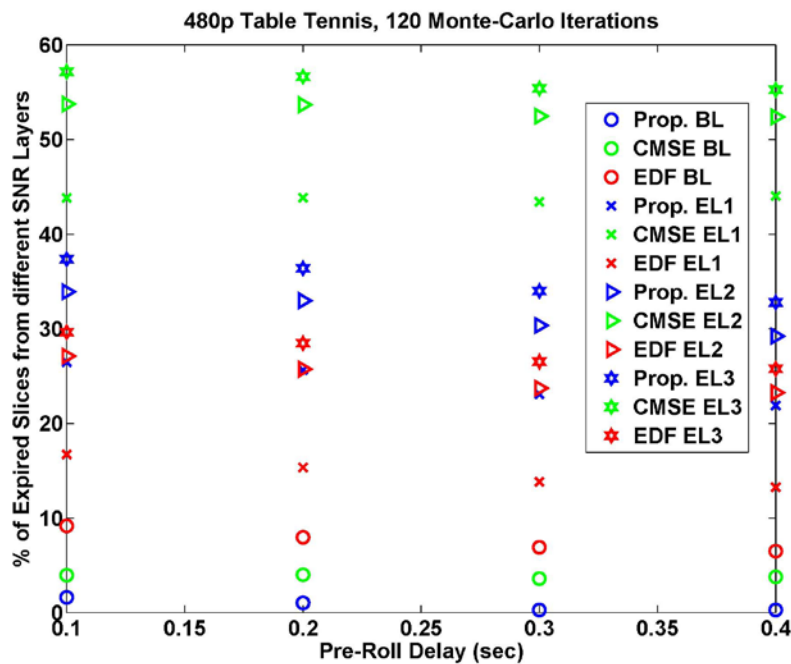


Figure 26: Total percentage of expired NAL units in 120 random channel realizations for EDF-based, CMSE-based, and proposed schemes.

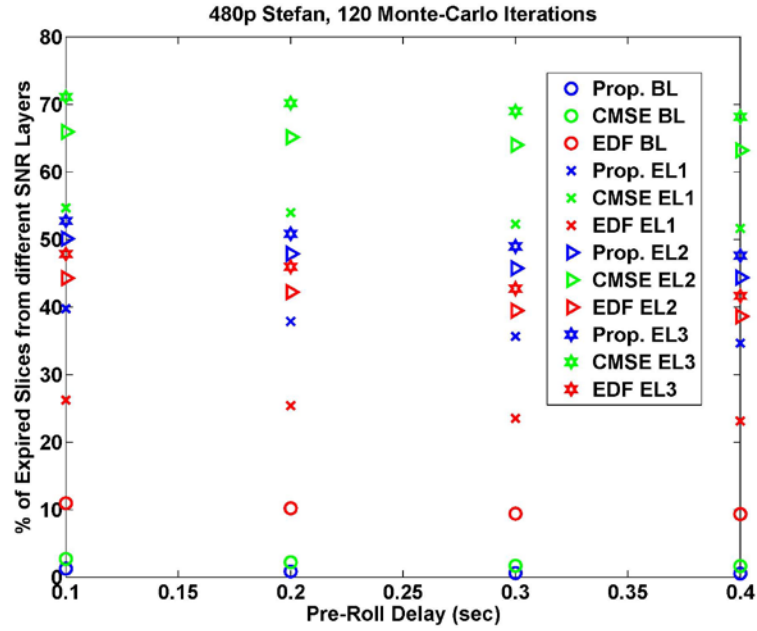
Figure 27 shows the percentage of expired NAL units belonging to different SNR layers in Table Tennis and Stefan. Here, the second, third, and fourth quality enhancement layers are denoted as EL1, EL2, and EL3. Our proposed scheme has significantly reduced the percentage of expired BL NAL units and hence, also significantly reduced the distortion caused by complete frame loss as compared to the EDF-based scheme. However, this is

achieved at the expense of more smaller-sized NAL units belonging to the higher quality enhancement layers. In the CMSE-based scheme, more NAL units in EL1, EL2, and EL3 expire than in our proposed scheme. This is because during every TTI the smaller-sized NAL units of higher SNR layers in the window fall behind in the scheduling order. For example, at a pre-roll delay of 0.2 seconds, almost 58% of NAL units in  $T_3$  expire in the CMSE-based scheme compared to 38% in our proposed scheme and 29% in the EDF-based scheme. The discarded NAL units belonging to the higher SNR layers EL1, EL2, and EL3 also include the events where they were discarded because the BL of that frame had expired. For example, at a pre-roll delay of 0.2 seconds for Table Tennis, 1% of the EL3 NAL units were discarded in our proposed scheme because the BL NAL units expired, an additional 25% were discarded when EL1 NAL units expired, and additional 8% were discarded when EL2 NAL units expired. Finally, only 3% had actually contended for a scheduling spot but failed.



(a)



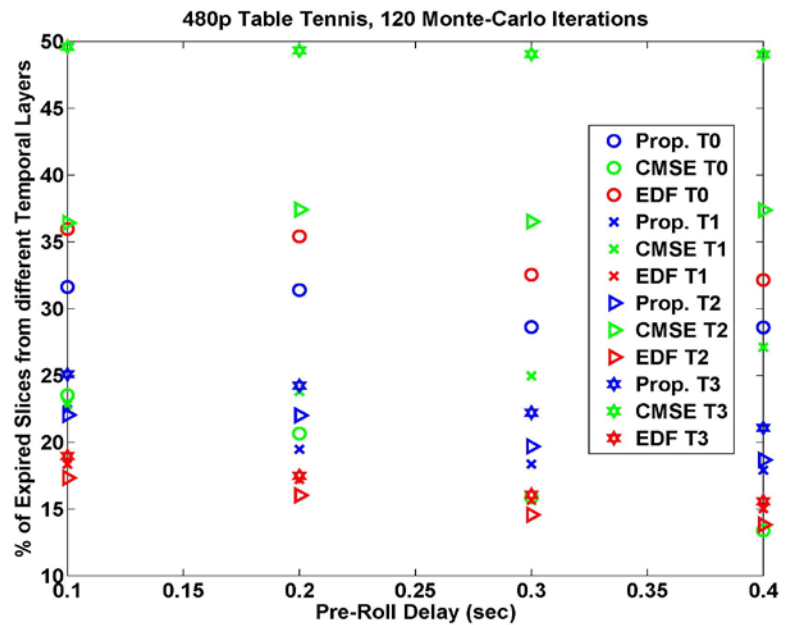


(b)

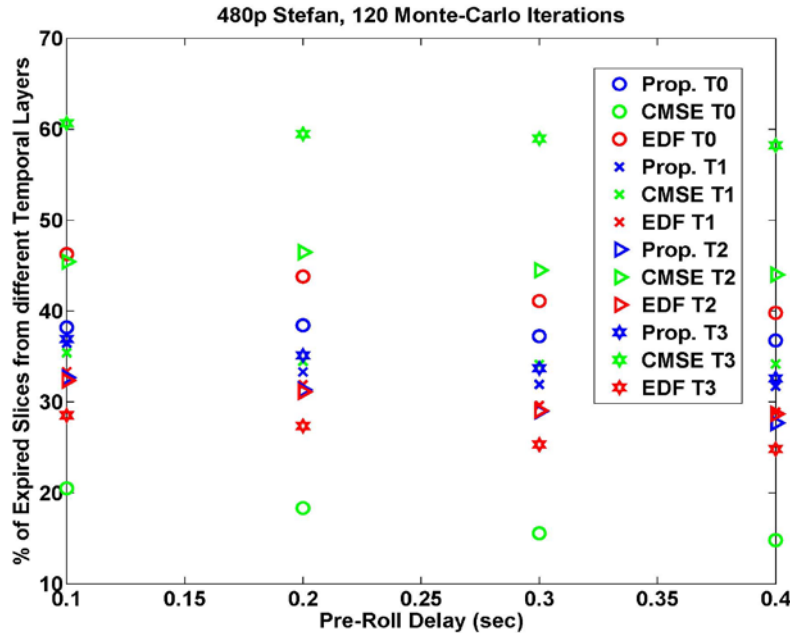
Figure 27: Percentage of expired NAL units in different SNR quality layers from 120 random channel realizations of EDF-based, CMSE-based, and proposed schemes.

Figures 28(a) and 28(b) show the percentage of expired NAL units from different temporal layers averaged over 120 random channel realizations. We observe that a greater percentage of NAL units expire from  $T_0$  in both the EDF-based and proposed schemes compared to the CMSE-based scheme. However, as shown in Figure 25 very few whole frames in  $T_0$  expire in all the three schemes, indicating that the expired NAL units belong to higher SNR layers of  $T_0$ . On the other hand, a higher percentage of NAL units expire from  $T_2$  and  $T_3$  temporal layers in the CMSE-based scheme. Figure 28 shows that for all temporal layers, the expired slices are comprised of few BL NAL units and significantly more NAL units belonging to the higher SNR layers. Tables 11 and 12 show that NAL units in  $T_0$  are much larger in size than the NAL units in  $T_1$ ,  $T_2$ , and  $T_3$  layers and therefore, require more time to be transmitted. Also from Table 13 their average normalized CMSE values are usually smaller compared to the NAL units in  $T_1$ ,  $T_2$ , and  $T_3$ . Overall, our proposed scheme

achieves a trade-off by discarding fewer frames from lower temporal layers and relatively more frames from higher temporal layers. Similarly, it discards fewer BL NAL units and relatively more NAL units from higher SNR layers.



(a)

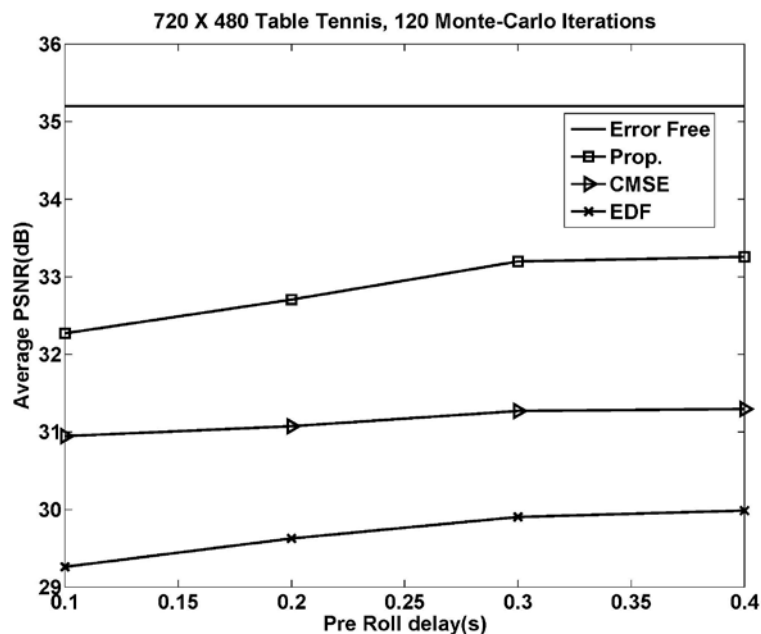


(b)

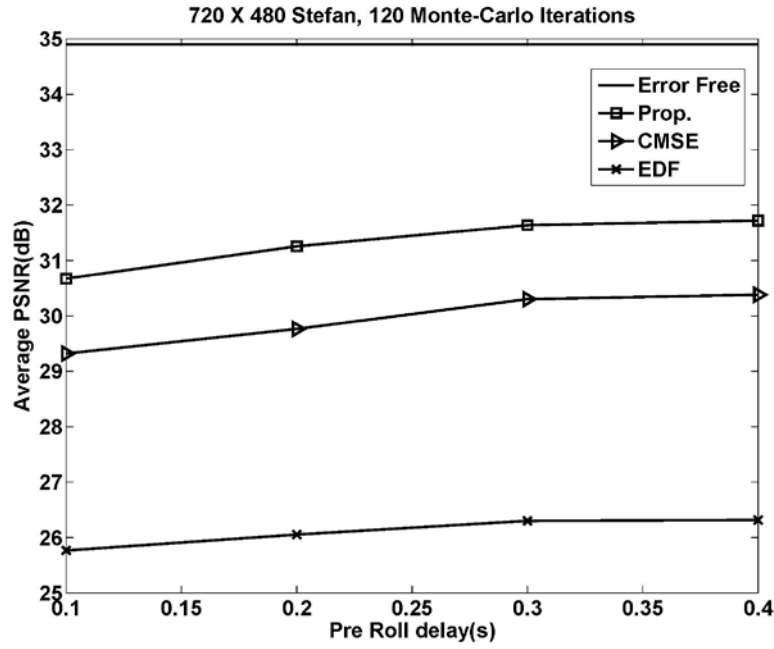
Figure 28: Percentage of expired NAL units in different temporal layers from 120 random channel realizations of EDF-based, CMSE-based, and proposed schemes.

#### 5.4.4 Evaluation of Video Quality

Figures 29(a) and 29(b) show the average video PSNR for Table Tennis and Stefan, computed over 120 different channel realizations for each pre-roll delay. Our proposed scheme achieves a PSNR gain of 3.3 dB (for Table Tennis) at pre-roll delays of 0.3 and 0.4 seconds and 5.4 dB (for Stefan) at a pre-roll delay of 0.4 seconds, over the EDF-based scheme. It also achieves PSNR gains of 2 dB (for Table Tennis) at pre-roll delays of 0.3 and 0.4 seconds, and 1.5 dB (for Stefan) at a pre-roll delay of 0.2 seconds, over the CMSE-based scheme. The poor video quality of the EDF-based scheme is primarily attributed to the whole video frames being discarded in close proximity. To illustrate this, we plot the frame to frame performance of the EDF-based and proposed schemes in one of the 120 channel realizations.



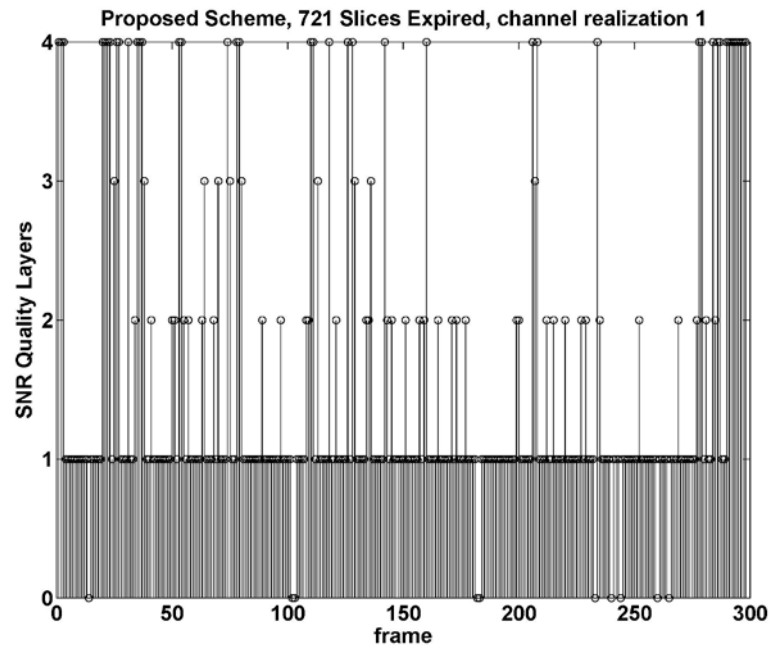
(a)



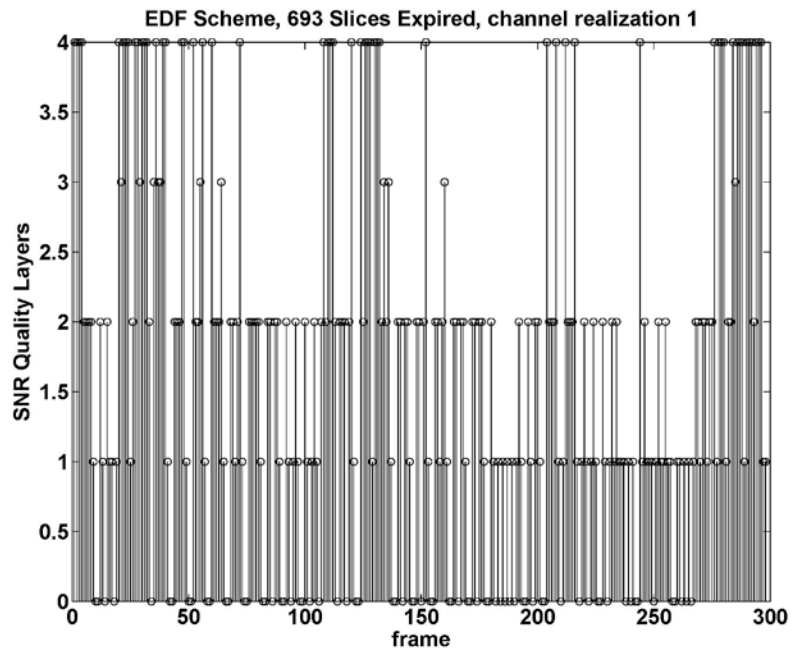
(b)

Figure 29: Average video PSNR of the EDF-based, CMSE-based, and proposed schemes over 120 random channel realizations.

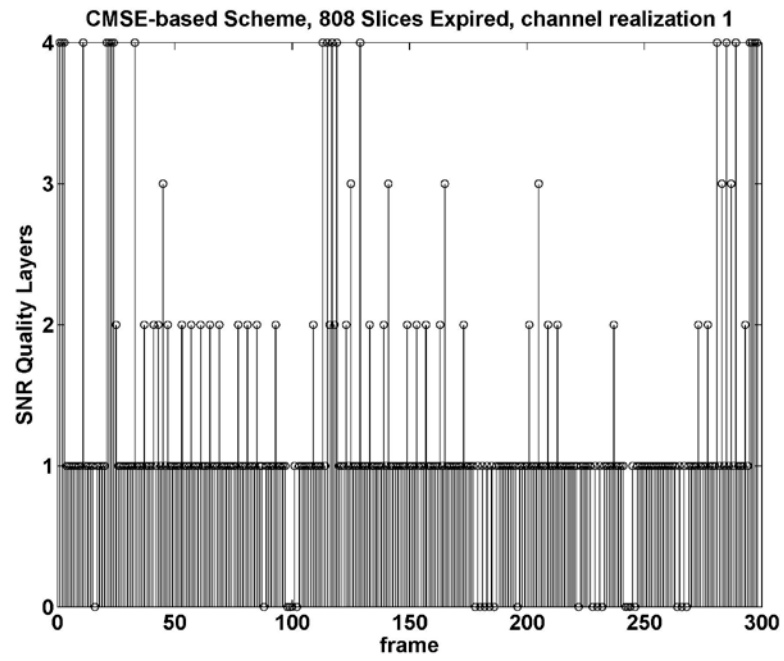
Figures 30(a), (b), and (c) show the number of SNR quality layers received for every video frame in the proposed, EDF-based, and CMSE-based schemes for a pre-roll delay of 0.1 seconds. Figures 31(a), (b), and (c) show the same for a pre-roll delay of 0.4 seconds. If the reference frames belonging to  $T_0$ ,  $T_1$ , and  $T_2$  layers are affected by expired NAL units, the distortion propagates to other frames in the GOP. When the number of SNR layers on the y-axis is zero, it indicates that the whole frame has expired. Frames which only play out the BL, show only one SNR quality layer on the y-axis.



(a)

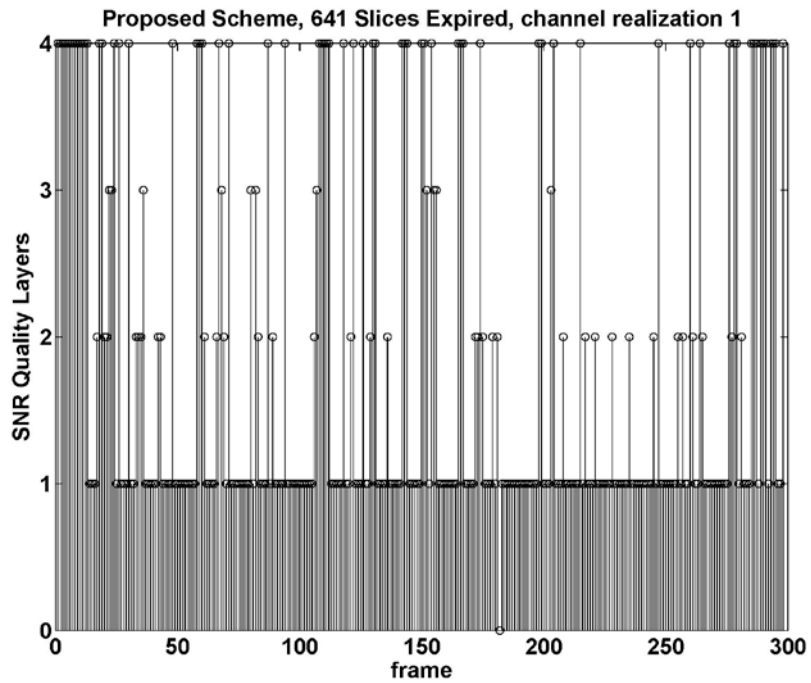


(b)

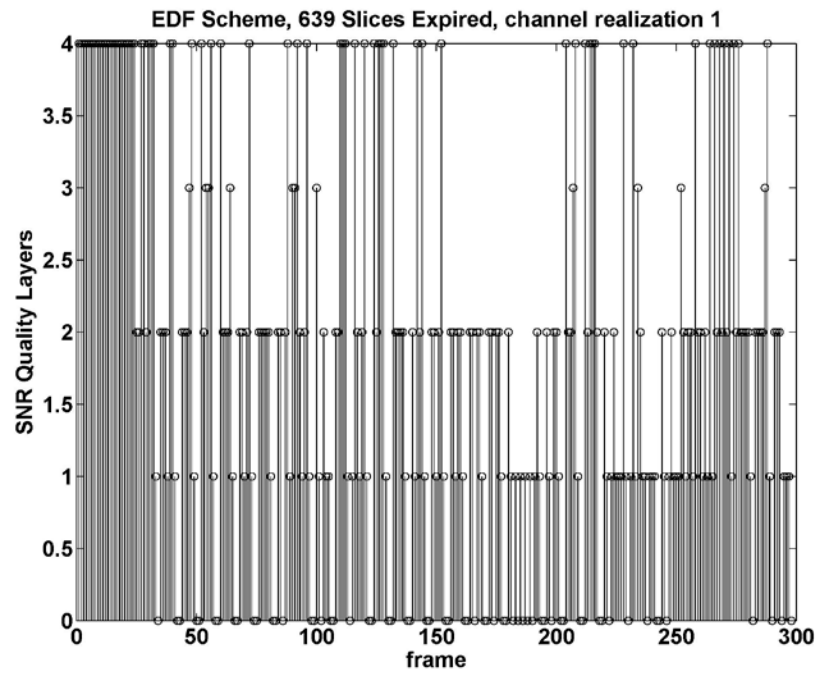


(c)

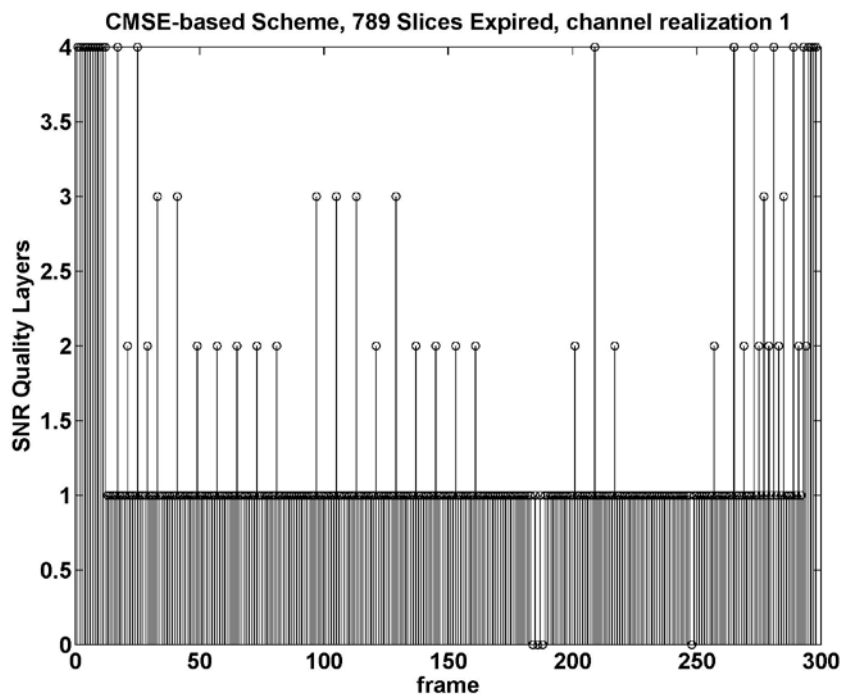
Figure 30: Per-frame video quality comparison between the proposed, EDF-based and CMSE-based schemes for Stefan at pre-roll delay of 0.1 s.



(a)



(b)



(c)

Figure 31: Per-frame video quality comparison between the proposed, EDF-based and CMSE-based schemes for Stefan at pre-roll delay of 0.4 s.

## 5.5 Conclusions

The EDF-based scheme dropped many frames in close proximity causing larger quality degradation. Also the EDF-based scheme showed large fluctuations in video quality, because some frames within the same GOP are completely discarded whereas some other frames have higher SNR layers scheduled. In our proposed scheme, for those GOPs in which complete frames were discarded, very few frames had their higher SNR layers scheduled. The CMSE-based scheme showed less fluctuation in video quality as compared to the EDF-based scheme but it still discarded some whole frames in close proximity. Finally, we also observed that as we go from a lower to higher pre-roll delay, more higher quality SNR layers are delivered for the frames.



## 6.0 CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

### 6.1 Conclusions

The military networks need to transmit time-sensitive information for situational awareness purpose. These networks may contain a constellation of hundreds of aircrafts and UAVs transporting high speed imagery and real-time collaborative voice and video. The wireless nodes should be capable of establishing connections with other node(s), whether airborne, in space, or on the surface, as needed. Lately, the full motion video has been widely adopted for situational awareness and surveillance.

The challenge in military networks is to organize a low-delay, reliable, infrastructure-less wireless network in the presence of highly dynamic network topology, heterogeneous nodes, intermittent transmission links and dynamic spectrum allocation. Most of these challenges are not present in commercial networks which enjoy better infrastructure and more predictable traffic. Existing cross-layer network protocols do not take a holistic view of these challenges and focus on one or a few aspects of the problem. The data representation and QoS-aware, cross-layer network protocols are *key enablers* in effectively deploying the military wireless networks. These network protocols should be closely integrated with the physical, data link and application layers. Specifically these protocols should consider the application and user QoS demands.

## 6.2 Contributions

Three cross-layer protocols were developed for the transmission of delay-sensitive and prioritized data in wireless networks. These protocols considered the QoS issues at different network layers. We used the H.264 compressed video packets as an example of the prioritized and delay-sensitive data. A novel cross-layer scheme was developed, which minimizes the expected received video distortion by jointly optimizing the packet sizes at the APP layer and estimating their FEC code rates to be allocated at PHY layer for the bit-rate limited and noisy channels. The optimization was carried out by considering the source bit rate, packet priority, latency, channel bandwidth, and SNR. To reduce the delays, the scheme was also extended to work on each video frame independently by predicting its expected channel bit budget. We then developed a cross-layer FEC scheme, which jointly optimizes the UEP Raptor codes at APP layer and UEP RCPC codes at PHY layer for the prioritized video packets, in order to minimize the distortion for the given source bit rates and channel constraints. Both the above schemes are novel because existing cross-layer schemes do not consider the influence of the packet priority in distortion minimization. Both schemes outperformed existing schemes in literature and provided significantly better video quality over bit-rate limited and lossy wireless channels.

Finally, a video slice CMSE and deadline-aware sliding-window based scheduling algorithm was designed, which exploits the temporal and SNR scalability of a H.264/SVC compressed bit stream for transmission over a wireless link with time-varying bit rate. This scheme effectively trades off the importance of the NAL units with their deadlines and determines a good transmission order for them. The proposed scheduling scheme reduced the whole frame losses by

taking into consideration the relative TTE of the NAL units, and thereby provided graceful degradation in bad channel conditions.

### **6.3 Future Research and Recommendations**

In this research, we did not consider the use of directional antennas, energy consumption, and dynamic spectrum allocation. The directional transmission can be attractive in military networks because it reduces the interference from other nodes, saves energy, increases network throughput, and reduces the threat of detection and jamming. Similarly, designing the energy efficient protocols is beneficial in increasing node and network life-time. However, the use of these factors would significantly affect the performance of existing QoS-aware, cross-layer protocols, and bring several new limiting factors to the design of current protocols. It would, therefore, be interesting to investigate their impact on the performance of existing cross-layer protocols. The insights can then be used in designing the novel QoS-aware, cross-layer protocols which are energy efficient and use the directional antennas.

## 7.0 REFERENCES

1. "Airborne Network Architecture," USAF Airborne Network Special Interest Group, Version 1.1, Oct. 7, 2004.
2. T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, July 2003.
3. H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 17, no. 9, pp. 1103–1120, September 2007.
4. S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan, "Error resiliency schemes in H.264/AVC standard," *Elsevier J. Vis. Commun. Image Rep.*, Special issue on Emerging H.264/AVC Video Coding Standard, vol. 17, no. 2, pp. 183–185, April 2006.
5. T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 657–673, July 2003.
6. Q. Zhang, W. Zhu, and Y.-Q. Zhang, "End-to-end QoS for video delivery over wireless internet," *Proc. IEEE*, vol. 93, no. 1, pp. 123–134, January 2005.
7. X. Zhu and B. Girod, "Video streaming over wireless networks," in *15th EUSIPCO*, Sept. 3–7, 2007, pp. 1462–1466.
8. Y. -M. Hsiao, J. -F. Lee, J. -S. Chen, and Y. -S. Chu, "H.264 video transmissions over wireless networks: Challenges and solutions," *Elsevier J. Comput. Commun.*, vol. 34, pp. 1661–1672, April 2011.
9. S. Shakkotai, T. Rappaport, and P. Karlsson, "Cross-layer design for wireless networks," *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 74–80, October 2003.
10. M. van der Schaar and N. S. Shankar, "Cross-layer wireless multimedia transmission: Challenges, principles, and new paradigms," *IEEE Wireless Commun. Mag.*, vol. 12, no. 4, pp. 50–58, August 2005.
11. E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod, "Cross-layer design of ad-hoc networks for real-time video streaming," *IEEE Wireless Commun.*, vol. 12, no. 4, pp. 50–58, 2005.
12. S. Kumar, V. S. Raghavan and J. Deng, "Medium Access Control Protocols for Ad-Hoc Wireless Networks: A Survey," *Elsevier Ad-Hoc Networks Journal*, Vol. 4(3), pp. 326–358, May 2006.
13. M. Sarkar, S. Gurajala and S. Kumar, "A QoS-Aware Medium Access Control Protocol for Real Time Traffic in Ad Hoc Networks," *18th IEEE Intl. Symp. Personal Indoor and Mobile Radio Communications (PIMRC'07)*, Athens, Greece, 3–7 Sept. 2007.

14. N. Farber and B. Girod, *Compressed Video over Networks*. Marcel Dekker, 2000, Chapter: Wireless Video, pp. 465–512.
15. I. F. Akyildiz et al., “NeXt generation / dynamic spectrum access/ cognitive radio wireless networks: A survey,” *Computer Networks*, vol. 50, pp. 2127-2159, 2006.
16. “XG Communications Program Overview,” DARPA Presentation by Preston Marshall and Todd Martin, WAND Industry Day Workshop, Feb. 27, 2007.
17. S. Chan, “Shared spectrum access for DOD,” *IEEE Commun. Mag.*, pp. 58-66, June 2007.
18. P. Kyasanur and N. H. Vaidya “Protocol design challenges for multi-hop dynamic spectrum access networks,” *Proc. IEEE DySPAN’05*, pp. 645-648, Nov. 8-11, 2005.
19. E. Modiano, “An adaptive algorithm for optimizing the packet size used in wireless ARQ,” *Wirel. Netw. J.*, vol. 5, no. 4, pp. 279 – 286, July 1999.
20. J. Korhonen and Y. Wang, “Effect of packet size on loss rate and delay in wireless links,” *IEEE WCNC*, vol. 3, 13 - 17 March 2005, pp. 1608 – 1613.
21. C.-W. Lee, C.-S. Yang, and Y.-C. Su, “Adaptive UEP and packet size assignment for scalable video transmission over burst-error channels,” *EURASIP J. App. Signal Process.*, vol. 2006, pp. 1 – 9, January 2006.
22. K. K. R. Kambhatla, S. Kumar, S. Paluri, and P. C. Cosman, “Wireless H.264 Video Quality Enhancement through Optimal Prioritized Packet Fragmentation,” *IEEE Trans. Multimedia*, vol. 14, no. 5, pp. 1480–1495, October 2012.
23. S. Paluri, K. K. R. Kambhatla, S. Kumar, B. Bailey, and P. Cosman, “Predicting slice loss distortion in H.264/AVC video for low complexity data prioritization,” *IEEE ICIP*, pages 689 – 692, Orlando, FL, USA, Sept. 30 - Oct. 3, 2012.
24. B. Y. Choi, S. Song, Y. Wang, and E. K. Park, “Using RTT variability for adaptive cross-layer approach to multimedia delivery in heterogeneous networks,” *IEEE Trans. Multimedia*, vol. 11, no. 6, pp. 1194 – 1203, October 2009.
25. C.-W. Lee, C.-S. Yang, and Y.-C. Su, “Low-complexity adaptive packet size assignment schemes for real-time scalable video transmission over WLANs,” *ACM IWCMC*, Leipzig, Germany, 21 - 24 June, 2009, pp. 1062 – 1066.
26. C.-H. Shih, “Adaptive forward error correction combined with packet size control for wireless video,” *IEEE 6th IHH-MSP*, Darmstadt, Germany, 15 - 17 October 2010, pp. 256 – 259.
27. H. Lin, T.-Y. Wu, and C.-Y. Huang, “Payload length adaptation for wireless video transmission in multicarrier systems,” in *IEEE VTC*, San Francisco, CA, USA, 5 - 8 September 2011, pp. 1 – 5.
28. M.-F. Tsai, N. Chilamkurti, and C.-K. Shieh, “An adaptive packet and block length forward error correction for video streaming over wireless networks,” *Springer Wireless Pers. Commun.*, vol. 56, no. 3, pp. 435 – 446, February 2011.
29. C.-H. Shih, “Enhancing packet-level forward error correction for streaming video in wireless networks,” *Intl. J. Computer Science Issues*, vol. 9, no. 5, pp. 146–155, September 2012.

30. T.-L. Lin and P. C. Cosman, "Efficient optimal RCPC code rate allocation with packet discarding for pre-encoded compressed video," *IEEE Signal Process. Lett.*, vol. 17, no. 5, pp. 505–508, May 2010.
31. T. Yoo, R. J. Lavery, A. Goldsmith, and D. J. Goodman, "Throughput optimization using adaptive techniques," *IEEE Commun. Lett.*, pp. 1–7, 2006.
32. S. Choudhury and J. D. Gibson, "Throughput optimization for wireless LANs in the presence of packet error rate constraints," *IEEE Commun. Lett.*, vol. 12, no. 1, pp. 11 – 13, January 2008.
33. T.-Y. Wu, T.-T. Chuang, and C. Y. Huang, "Optimal transmission of high definition video transmission in wimedia systems," *Wirel. Netw. J.*, vol. 17, no. 2, pp. 291 – 303, February 2011.
34. S. Choudhury and J. D. Gibson, "Payload length and rate adaptation for multimedia communications in wireless LANs," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 4, pp. 796 – 807, May 2007.
35. H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, AC-19(6):716–723, 1974.
36. H. Akaike, Canonical correlation analysis of time series and the use of an information criterion. *R. K. Mehra and D. G. Lainotis (eds.), System Identification: Advances and Case Studies, Academic Press, New York*, pages 52–107, 1976.
37. S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ. Press, Cambridge, U.K., 2004.
38. J. Chakarakesi and P. A. Chou, "Application layer error-correction coding for rate-distortion optimized streaming to wireless clients," *IEEE Trans. Commun.*, 52(10):1675–1687, October 2004.
39. J. Hagenauer, "Rate-compatible punctured convolutinal codes (RCPC Codes) and their applications," *IEEE Trans. Commun.*, 36(4):389–400, April 1988.
40. E. J. Hannan and B. G. Quinn, "The determination of the order of an autoregression," *J. Royal Stat. Soc., B*, 41(2):190–195, 1979.
41. D. W. Hosmer and S. Lemeshow. *Applied Logistic Regression*. Probability and Statistics. John Wiley & Sons, 2nd edition, 2000.
42. D. Li and X. Sun. *Nonlinear Integer Programming*. International Series in Operations Research and Management Science, Springer, USA, 2006.
43. F. Li and G. Liu, "Compressed-domain-based transmission distortion modeling for precoded H.264/AVC video," *IEEE Trans. Circuits Syst. Video Technol.*, 19(12):1908 – 1914, December 2009.
44. H. Luo, S. Ci, and D. Wu, "A cross-layer design for the performance improvement of real-time video transmission of secondary users over cognitive radio networks," *IEEE Trans. Circuits Syst. Video Technol.*, 21(8):1040 – 1048, August 2011.
45. P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, 2nd edition, 1989.

46. M. H. Pinson and S. Wolf, "A new standardization method for objectively measuring video quality," *IEEE Trans. Broadcast.*, 50(3):312–322, 2004.
47. G. Schwarz, "Estimating the dimension of a model," *The Ann. Stat.*, 6(2):461–464, 1978.
48. A. Vosoughi, V. Testoni, P. Cosman, and L. Milstein, "Joint source-channel coding of 3D video using multiview coding," *IEEE ICASSP*, pages 2050 – 2054, 2013.
49. S. Wolf and M. H. Pinson, "Video quality measurement techniques," Technical report, National Telecommunications and Information Administration (NTIA), U.S. Department of Commerce, June 2002.
50. Y. Z. Huang and John G. Apostolopoulos, "A joint packet selection/omission and FEC system for streaming video," *IEEE ICASSP*, pages I845 - I848, Honolulu, HI, 15 - 20 April 2007.
51. RoHC: Robust Header Compression, RFC:3095 Std., July 2001.
52. Local and Metropolitan Area Networks-specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Standard for Information Technology-Telecommunications and Information Exchange between Systems Std., June 2007.
53. The R Project for Statistical Computing. Vienna, Austria. [Online]. Available: <http://www.r-project.org/>
54. T. Stockhammer, A. Shokrollahi, M. Watson, M. Luby, and T. Gasiba, "Application Layer Forward Error Correction for Mobile Multimedia Broadcasting," *Handbook of Mobile Broadcasting: DVB-H, DMB, ISDB-T and Media FLO*, Mar. 2008, pp. 239–280.
55. A. Talari, S. Kumar, N. Rahnavard, S. Paluri, and J. D. Matyjas, "Optimized Cross-layer Forward Error Correction Coding for H.264 AVC Video Transmission over Wireless Channels," *EURASIP J. Wireless Commun. and Networking*, Aug. 2013.
56. E. Baccaglioni, T. Tillo, and G. Olmo, "Slice Sorting for Unequal Loss Protection of Video Streams," *IEEE Signal Processing Letters*, vol. 15, Aug. 2008, pp. 581–584.
57. W. Xiang, C. Zhu, C. K. Siew, Y. Xu, and M. Liu, "Forward Error Correction-Based 2-D Layered Multiple Description Coding for Error-Resilient H.264/SVC Video Transmission," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, Dec. 2009, pp. 1730–1738.
58. D. G.- Barquero and A. Bria, "Application Layer FEC for Improved Mobile Reception of DVB-H Streaming Services," *IEEE 64th Vehicular Tech. Conf.*, Fall 2006, pp. 1–5.
59. T. Courtade and R. Wesel, "A Cross-Layer Perspective on Rateless Coding for Wireless Channels," *Proc. ICC*, Jun. 2009, pp. 1–6.
60. M. Luby, M. Watson, T. Gasiba, and T. Stockhammer, "Mobile Data Broadcasting over MBMS Tradeoffs in Forward Error Correction," *Proc. 5th Int. Conf. Mobile and Ubiquitous Multimedia*, Dec. 2006.
61. A. Alexiou, C. Bouras, and A. Papazois, "A Study of Forward Error Correction for Mobile Multicast," *Int. J. Commun. Systems*, vol. 24, 2011, pp. 607–627.

62. D. Munaretto, D. Jurca, and J. Widmer, "Broadcast Video Streaming in Cellular Networks: An Adaptation Framework for Channel, Video and AL-FEC Rates Allocation," Wireless Internet Conference (WICON), The 5th Annual ICST, 2010, pp. 1–9.
63. C. Bouras, N. Kanakis, V. Kokkinos, and A. Papazois, "Application Layer Forward Error Correction for Multicast Streaming over LTE Networks," International J. Communication Systems, 2012.
64. A. Shokrollahi, "Raptor Codes," IEEE Trans. Inf. Theory, vol. 52, Jun. 2006, pp. 2551–2567.
65. A. Shokrollahi and M. Luby, "Raptor Codes - Foundations and Trends in Communications and Information Theory," Foundations and Trends in Comm. and Inf. Theory, vol. 6, 2011, pp. 213–322.
66. 3GPP TS 26.346 V7.4.1, Technical Spec. Group Serv. and Sys. Aspects, "Multimedia Broadcast/Multicast Service (MBMS); Protocols and Codecs," 3GPP Technical Spec., Jun. 2007.
67. S. Ahmad, R. Hamzaoui, and M. Al-Akaidi, "Adaptive Unicast Video Streaming with Rateless Codes and Feedback," IEEE Trans. Circuits Syst. Video Technol., vol. 20, Feb. 2010, pp. 275–285.
68. P. Cataldi, M. Grangetto, T. Tillo, E. Magli, and G. Olmo, "Sliding-Window Raptor Codes for Efficient Scalable Wireless Video Broadcasting with Unequal Loss Protection," IEEE Trans. Image Processing, vol. 19, Jun. 2010, pp. 1491–1503.
69. Z. Cheng, J. Castura, and Y. Mao, "On the Design of Raptor Codes for Binary-Input Gaussian Channels," TCOMM, vol. 57, Nov. 2009, pp. 3269–3277.
70. K. Kambhatla, S. Kumar, P. Cosman, and J. D. Matyjas, "H.264/AVC Video Packet Aggregation and Unequal Error Protection for Noisy Channels," IEEE ICIP, Orlando, FL, Sept.-Oct. 2012.
71. A. A. Khalek, C. Caramanis, and R. W. H. Jr., "A Cross-Layer Design for Perceptual Optimization Of H.264/SVC with Unequal Error Protection," IEEE J. Selected Areas Comm., vol. 30, Aug. 2012, pp. 1157–1171.
72. T. A. Courtade and R. D. Wesel, "Optimal Allocation of Redundancy between Packet-Level Erasure Coding and Physical-Layer Channel Coding in Fading Channels," IEEE Trans. Commun., vol. 59, Aug. 2011, pp. 2101–2109.
73. F. Zhai, Y. Eisenberg, T. Pappas, R. Berry, and A. Katsaggelos, "Rate-Distortion Optimized Product Code Forward Error Correction for Video Transmission over IP-Based Wireless Networks," Proc. Int. Conf. Acoustics, Speech and Signal Processing (ICASSP2004), Montreal, Canada, Jun. 2004, pp. 857–860.
74. M. Luby, "LT Codes," Annual IEEE Symp. Found. of Comp. Sci., Nov. 2002, pp. 271–280.
75. Digital Video Broadcasting (DVB): Transmission System for Handheld Terminals (DVB-H), 2004.
76. H. Kushwaha, Y. Xing, R. Chandramouli, and H. Heffes, "Reliable Multimedia Transmission over Cognitive Radio Networks using Fountain Codes," Proc. of the IEEE, vol. 96, Jan. 2008, pp. 155–165.



77. A. S. Tan, A. Aksay, C. Bilen, G. B. Akar, and E. Arikan, "Rate-Distortion Optimized Layered Stereoscopic Video Streaming with Raptor Codes," IEEE Intl. Conf. Multimedia and Expo, Nov. 2007, pp. 98–104.
78. D. Vukobratovic, V. Stankovic, D. Sejdinovic, L. Stankovic, and Z. Xiong, "Scalable Video Multicast using Expanding Window Fountain Codes," IEEE Trans. Multimedia, vol. 11, Oct. 2009, pp. 1094–1104.
79. T.-W. A. Lee, S.-H. G. Chan, Q. Zhang, W.-W. Zhu, and Y.-Q. Zhang, "Optimal Allocation of Packet-Level and Byte-Level FEC in Video Multicasting over Wired and Wireless Networks," Proc. GLOBECOM, Nov. 2001, pp. 1994–1998.
80. Q. Zhang, W.-W. Zhu, and Y.-Q. Zhang, "Channel-Adaptive Resource Allocation for Scalable Video Transmission over 3G Wireless Network," IEEE Trans. Circuits Syst. Video Technol., vol. 14, Aug. 2004, pp. 1049–1063.
81. N. Thomos, S. Argyropoulos, N. Boulgouris, and M. Strintzis, "Robust Transmission of H.264/AVC Streams using Adaptive Group Slicing and Unequal Error Protection," EURASIP J. Applied Signal Processing, 2006, pp. 120–120.
82. J. Proakis, Digital Communications, 3rd ed., McGraw Hill, New York, 1995.
83. P. Koopman and T. Chakravarty, "Cyclic Redundancy Code (CRC) Polynomial Selection for Embedded Networks," Int. Conf. Dependable Systems and Networks, Jul. 2004, pp. 145–154.
84. E. Maani and A. Katsaggelos, "Unequal Error Protection for Robust Streaming of Scalable Video over Packet Lossy Networks," IEEE Trans. Circuits Syst. Video Technol., vol. 20, Mar. 2010, pp. 407–416.
85. H. Ha and C. Yim, "Layer-Weighted Unequal Error Protection for Scalable Video Coding Extension of H.264/AVC," IEEE Trans. Consumer Electronics, vol. 54, May 2008, pp. 736–744.
86. J. Afzal, T. Stockhammer, T. Gasiba, and W. Xu, "Video Streaming over MBMS: A System Design Approach," J. Multimedia, vol. 1, 2006, pp. 25–35.
87. Mathworks, "Global Optimization Toolbox: Users Guide (r2011b)," Nov. 2011.
88. J. Holland, Adaptation in Natural and Artificial Systems. MIT press Cambridge, MA, 1992.
89. J. R. Koza, "Survey of Genetic Algorithms and Genetic Programming," WESCON Conference Record, Nov. 1995, p. 589.
90. D. Coley, An Introduction to Genetic Algorithms for Scientists and Engineers. World Scientific Pub Co Inc., 1999.
91. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II," IEEE Trans. Evolutionary Computation, vol. 6, Apr. 2002, pp. 182–197.
92. JVT, H.264/AVC reference software JM14.2, ISO/IEC Std. [Online]. Available: <http://iphome.hhi.de/suehring/tml/download/>.

93. O. Alay, T. Korakis, Y. Wang, and S. Panwar, "Dynamic Rate and FEC Adaptation for Video Multicast in Multi-Rate Wireless Networks," *J. Mobile Networks and Applications*, vol. 15, Jun. 2010, pp. 425–434.
94. R. Clarke, "A Statistical Theory of Mobile-Radio Reception," *Bell Syst. Tech.J.*, vol. 47, Jul. 1968, pp. 57–1000.
95. W. Jakes, *Microwave Mobile Communications*. New York: Wiley, 1974.
96. F. Zhai, Y. Eisenberg, T. Pappas, R. Berry, and A. Katsaggelos, "Rayleigh Fading Channels in Mobile Digital Communications: Parts I and II," *IEEE Commun. Mag.*, vol. 35, Jul. 1997, pp. 90–110.
97. J. R. Ohm, "Advances in scalable video coding," *Proc. IEEE*, vol. 93, no. 1, pp. 42–56, January 2005.
98. Y. Wang and S. Lin, "Error-resilient video coding using multiple description motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 438–452, August 2002.
99. C. I. Lin, W. N. Lie, and C. W. Lin, "Enhancing video error resilience by using data embedding techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 2, pp. 300–308, February 2006.
100. A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: An overview," *IEEE Signal Process. Mag.*, vol. 20, no. 2, pp. 18–29, March 2003.
101. C. W. Lin, J. Xin, and M. T. Sun, "Digital video transcoding," *Proc. IEEE*, vol. 93, no. 1, pp. 84–97, June 2005.
102. P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 390–404, April 2006.
103. M. Kalman, E. Steinbach, and B. Girod, "Adaptive media playout for low-delay streaming over error-prone channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 6, pp. 841–851, June 2006.
104. Y. F. Su, Y. H. Yang, and H. Chen, "Smooth control of adaptive media playout for video streaming," *IEEE Trans. Multimedia*, vol. 11, no. 7, pp. 1331–1339, November 2009.
105. Y. Li, A. Markopoulou, J. Apostolopoulos, and N. Bambos, "Content-aware playout and packet scheduling for video streaming over wireless links," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 885–895, August 2008.
106. H. Chiang, C. Y. Huang, and T. Chiang, "Content-aware adaptive media playout controls for wireless video streaming," *IEEE Trans. Multimedia*, vol. 9, no. 6, pp. 1273–1283, September 2007.
107. X. Liu, E. K. P. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks," *Elsevier J. Comput. Netw.*, vol. 41, no. 4, pp. 451–474, March 2003.
108. L. Georgiadis, M. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–144, 2006.

109. B. Girod, J. Chakareski, M. Kalman, Y. J. Liang, E. Setton, and R. Zhang, "Advances in network-adaptive video streaming," IWDC 2002, Capri, Italy, September 2002.
110. B. Girod, M. Kalman, Y. J. Liang, and R. Zhang, "Advances in channel-adaptive video streaming," IEEE ICIP, Rochester, NY, September 2002.
111. M. Kalman and B. Girod, *Multimedia Networking and Communication. Network-Adaptive Media Streaming*, Elsevier, pp. 223–232, 2007.
112. N. G. Duffield, K. K. Ramakrishnan, and A. R. Reibman, "SAVE: an algorithm for smoothed adaptive video over explicit rate networks," IEEE INFOCOM, San Francisco, CA, USA, March 1998, pp. 1093–1102.
113. T. Ott, T. V. Lakshman, and A. Tabatabai, "A scheme for smoothing delay sensitive traffic offered to ATM networks," IEEE INFOCOM, Florence, Italy, 4-8 May 1992, pp. 776–785.
114. Z. L. Zhang, J. Kurose, J. Salehi, and D. Towsley, "Smoothing, statistical multiplexing, and call admission control for stored video," IEEE J. Sel. Areas Commun., vol. 15, no. 6, pp. 1148–1166, August 1997.
115. M. Kalman, E. Steinbach, and B. Girod, "Adaptive media playout for low delay video streaming over error-prone channels," IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 6, pp. 841–851, June 2004.
116. M. Kalman and B. Girod, "Rate-distortion optimized video streaming with multiple deadlines for low latency applications," Packet Video Workshop, Irvine, CA, December 2004.
117. J. Chakareski, J. G. Apostolopoulos, S. Wee, W. T. Tan, and B. Girod, "Rate-distortion hint tracks for adaptive video streaming," IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 10, pp. 1257–1269, October 2005.
118. T. Stockhammer, H. Jenkac, and G. Kuhn, "Streaming video over variable bit-rate wireless channels," IEEE Trans. Multimedia, vol. 6, no. 2, pp. 268–277, April 2004.
119. Z. Chen and Y. Reznik, "Analysis of video codec buffer and delay under time-varying channel," IEEE VCIP, 27-30 Nov. 2012, pp. 1–6.
120. Q. Zhang and S. A. Kassam, "Finite-state Markov model for Rayleigh fading channels," IEEE Trans. Commun., vol. 47, no. 11, pp. 1688 – 1692, November 1999.
121. P. Sadeghi, R. A. Kennedy, P. B. Rapajic, and R. Shams, "Finite-state Markov modeling of fading channels," IEEE Signal Process. Mag., vol. 25, no. 5, pp. 57 – 80, September 2008.
122. H. C. Chuang, C. Y. Huang, and T. H. Chiang, "Content-aware adaptive media playout controls for wireless video streaming," IEEE Trans. Multimedia, vol. 9, no. 6, pp. 1273 – 1283, Sept. 2007.
123. Y. F. Su, Y. H. Yang, M. T. Lu, and H. H. Chen, "Smooth control of adaptive media playout for video streaming," IEEE Trans. Multimedia, vol. 11, no. 7, pp. 1331 – 1339, Nov. 2009.
124. S. H. Kang and A. Zakhori, "Packet scheduling algorithm for wireless video streaming," International Packet Video Workshop, 2002, pp. 1–11.

125. Z. Jiang and L. Kleinrock, "A packet selection algorithm for adaptive transmission of a smoothed video over a wireless channel," *Elsevier J. Parallel Distr. Com.*, vol. 60, no. 4, pp. 494–509, April 2000.
126. T. Schierl, Y. S. de la Fuente, R. Globisch, C. Hellge, and T. Wiegand, "Priority-based media delivery using SVC with RTP and HTTP streaming," *Springer Multimed. Tools Appl.*, vol. 55, no. 2, pp. 227–246, Sept. 2011.
127. S. Xiao, H. Wang, and C.-C. J. Kuo, "A practical bit stream organization algorithm for robust H.264/SVC transmission," *Elsevier J. Vis. Commun. and Image Rep.*, vol. 21, no. 8, pp. 871–879, November 2010.
128. Z. Chen and D. Wu, "Rate-distortion optimized cross-layer rate control in wireless video communication," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 3, pp. 352–365, March 2012.
129. Z. He and S. K. Mitra, "Optimum bit allocation and accurate rate control for video coding p-domain source modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 10, pp. 840–849, October 2002.
130. A. Dua, C. W. Chan, N. Bambos, and J. Apostolopoulos, "Channel, deadline and distortion (CD2) aware scheduling for video streams over wireless," *IEEE Trans. Wireless Commun.*, vol. 9, no. 3, pp. 1001–1011, March 2010.
131. A. Comberoux, C. Delestre, N. Changuel, B. Sayadi, and M. Kieffer, "Cross-layer optimization of a multimedia streaming system via dynamic programming," *IEEE ICIP*, Sept. 30 - Oct. 3 2012, pp. 2261–2264.
132. Y. Zhang, F. Fu, and M. van der Schaar, "On-line learning and optimization for wireless video transmission," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3108–3124, June 2010.
133. Advanced Video Coding for Generic Audio Visual Services, ITU-T H.264 - ISO/IEC 14496-10 Std., January 2012.
134. JSVM 9.8. JSVM SVC reference software manual. [Online]. Available: <http://ube.ege.edu.tr/~boztok/JSVM/SoftwareManual.pdf>.
135. Y. Sanchez, T. Schierl, C. Hellge, T. Wiegand, D. Hong, D. D. Vleeschauwer, W. V. Leekwijck, and Y. L. Louedec, "Efficient HTTP-based streaming using scalable video coding," *Elsevier Signal Process. Image*, vol. 27, no. 4, pp. 329–342, April 2012.
136. J. McDougall and S. Miller, "Sensitivity of wireless network simulations to a two-state Markov model channel approximation," *IEEE Globecom*, vol. 2, San Francisco, CA, Dec. 1-5 2003, pp. 697–701.
137. S. Karande, M. K. S. Khayam, and H. Radha, "Analysis and modeling of errors at the 802.11b link layer," in *IEEE ICME*, vol. 1, 6-9 July 2003, pp. 673–676.
138. J. Hartwell and A. Fapojuwo, "Modeling and characterization of frame loss process in IEEE 802.11 wireless local area networks," *IEEE VTC '04*, vol. 6, Los Angeles, CA, Sept. 26-29 2004, pp. 4481–4485.
139. S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementation*. John Wiley and Sons, 1990.

140. L. Caccetta and A. Kulanoot, "Computational aspects of hard Knapsack problems," *Nonlinear Anal. - Theor.*, vol. 47, no. 8, pp. 5547–5558, 2001.
141. A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, W. Stephen, Ed. McGraw-Hill Science/Engineering/Math, December 2001.
142. L. Georgiadis, R. Guerin, and A. Parekh, "Optimal multiplexing on a single link: Delay and buffer requirements," *IEEE/ACM Trans. Netw.*, vol. 43, no. 5, pp. 1518–1535, Sept. 1997.
143. A. A. Khalek, C. Caramanis, and R. W. Heath, "Joint source-channel adaptation for perceptually optimized scalable video transmission," *IEEE Globecom*, Houston, TX, December 2011, pp. 1–5.
144. T.-Y. Hung, Z. Chen, and Y.-P. Tan, "Playout adaptation based packet scheduling for scalable video delivery over wireless networks," *IEEE ICME*, 19-23 July 2010, pp. 1022–1027.
145. C. Chen, R. W. Heath, A. C. Bovik, and G. de Veciana, "Adaptive policies for real-time video transmission: A Markov decision process framework," *IEEE ICIP*, 11-14 September 2011, pp. 2249–2252.

## LIST OF ACRONYMS

AF	Air Force
AIC	Akaike Information Criterion
AODV	Ad Hoc On-demand Distance Vector Routing
AOMDV	Ad Hoc On-demand Multipath Distance Vector Routing
APP	Application Layer
ARQ	Automatic Repeat Request
AVC	Advanced Video Coding
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BL	Base Layer
BPSK	Binary Phase Shift Keying
BSC	Binary Symmetric Channel
CBR	Constant Bit Rate
CIF	Common Intermediate Format
CMSE	Cumulative Mean Squared Error
CRC	Cyclic Redundancy Check
CSI	Channel Side Information
DP	Dynamic Programing
DVB	Digital Video Broadcast
EDF	Earliest Deadline First
EEP	Equal Error Protection
EL	Enhancement Layer

FEC	Forward error correction
FMO	Flexible Macroblock Ordering
GA	Genetic Algorithm
GLM	Generalized Linear Model
GOP	Group of Pictures
HQIC	Hannan-Quinn Information Criterion
HTTP	Hypertext Transfer Protocol
IDR	Instantaneous Decoding Refresh
IMSE	Initial Mean Squared Error
JSVM	Joint Scalable Video Model
LAN	Local Area Network
LTE	Long Term Evolution
LT	Luby Transform
MAC	Medium Access Layer
MB	Macroblock
MANET	Mobile Ad-hoc Network
MDP	Markov Decision Process
MGS	Medium Grain Scalability
MSE	Mean Squared Error
MTU	Maximum Transmission Unit
NAL	Network Abstraction Layer
OCRA	Optimal Code Rate Allocation
PER	Packet Error Rate

PHY	Physical Layer
PSNR	Peak Signal to Noise Ratio
QoS	Quality of Service
RCPC	Rate Compatible Punctured Convolutional codes
RD or R-D	Rate Distortion
RoHC	Robust Header Compression
RTP	Real-time Transport Protocol
SEI	Supplemental Enhancement Information
SNR	Signal to Noise Ratio
TTE	Time to Expiry
TTI	Transmission Time Interval
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UEP	Unequal Error Protection
VBR	Variable Bit Rate
VCL	Video Coding Layer
VQM	Video Quality Metric
VSLR	Video Slice Loss Ratio
3GPP	Third Generation Partnership Project